# BUSINESS APPLICATIONS for the Apple II&IIe under CP/M

Steven Zimmerman
Leo Conrad
Donald Smith

# Business Applications for the Apple II and IIe Under CP/M

Publishing Director: David Culverwell
Acquisitions Editor: Terrell Anderson
Production Editor: Len Blasso
Text Designer: Paula Huber
Art Director/Cover Design: Don Sellers
Assistant Art Director: Bernard Vervin
Manufacturing Director: John Komsa

Indexer: Elliot Linzer
Typesetter: Techna Type, York, PA
Printer: Fairfield Graphics, Fairfield, PA

# BUSINESS APPLICATIONS for the APPLE II and IIe Under CP/M

*Steven M. Zimmerman, Ph.D.*
*Leo M. Conrad*
*Donald R. Smith, Ph.D., P.E.*

Brady Communications Company, Inc., Bowie, MD 20715
A Prentice-Hall Publishing Company

Business Applications for the Apple II and IIe Under CP/M

Copyright © 1985 by Brady Communications Company, Inc.
All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, or by any information storage and retrieval system, without permission in writing from the publisher. For information, address Brady Communications Company, Inc., Bowie, Maryland 20715.

Library of Congress Cataloging in Publication Data

Zimmerman, Steven M.
Business applications for the Apple II and IIe under CP/M.

Includes index.
1. Apple II (Computer)—Programming. 2. Apple IIe (Computer)—Programming. 3. CP/M (Computer operating system). 4. Business—Data processing. I. Conrad, Leo M. II. Smith, Donald R. III. Title.
HF5548.4.A65Z56 1984 658'.022'0285425 84-20496
ISBN 0-89303-354-5

Prentice-Hall International, Inc., *London*
Prentice-Hall Canada, Inc., Scarborough, *Ontario*
Prentice-Hall of Australia, Pty., Ltd., *Sydney*
Prentice-Hall of India Private Limited, *New Delhi*
Prentice-Hall of Japan, Inc., *Tokyo*
Prentice-Hall of Southeast Asia Pte. Ltd., *Singapore*
Whitehall Books, Limited, Petone, *New Zealand*
Editora Prentice-Hall Do Brasil LTDA., *Rio de Janeiro*
Prentice-Hall Hispanoamericana, S.A., *Mexico*

Printed in the United States of America

85 86 87 88 89 90 91 92 93 94 95 1 2 3 4 5 6 7 8 9 10

# CONTENTS

Preface

PART I The Apple II and Its Systems

| | | |
|---|---|---|
| 1 | The Apple II Microcomputer System with CP/M | 3 |
| 1.1 | What Does It Look Like? | 3 |
| 1.2 | What Is It? | 5 |
| 1.3 | Software To Run The Apple II System | 6 |
| 1.4 | Examining Some Add-Ons | 7 |
| 1.5 | Setting Up Your System | 9 |
| 1.6 | Getting Started | 9 |
| 1.7 | Printer Graphics | 11 |
| 1.8 | Copying Programs | 11 |
| 1.9 | For Dual Drive Systems | 12 |
| 1.10 | For Single Disk Drive System | 15 |
| 1.11 | Summary | 18 |
| 2 | Apple CP/M | 20 |
| 2.1 | Your Computer's Hardware | 21 |
| 2.2 | Getting Your Operating System Started | 22 |
| 2.3 | A Look At Data Storage On Disks | 23 |
| 2.4 | Starting Apple CP/M | 26 |
| 2.5 | Apple-CP/M Commands | 27 |
| 2.6 | CP/M File Name Extents | 30 |
| 2.7 | Summary | 31 |

PART II Accounting

| | | |
|---|---|---|
| 3 | A Collection Of Business Programs | 35 |
| 3.1 | Accounting Knowledge | 37 |
| 3.2 | Organization Of This Section | 38 |
| 3.3 | System Overview | 39 |
| 3.4 | RUN "x:MENU" | 41 |
| 3.5 | Examining The Program | 42 |
| 3.6 | Summary | 43 |
| 3.7 | Program Listing | 43 |
| 4 | (S)etup Chart Of Accounts | 45 |
| 4.1 | Some Accounting Background | 46 |
| 4.2 | Overview Of Numbering System For Chart Of Accounts | 46 |
| 4.3 | Running The Program | 47 |

| | | |
|---|---|---|
| 4.4 | A Second Run to Input FIRST.BAL And FIRST.INC | 52 |
| 4.5 | Completing the file FIRST.INC | 55 |
| 4.6 | Some Additional Abilities | 58 |
| 4.7 | The Other Files | 62 |
| 4.8 | (N)umbers | 63 |
| 4.9 | Examining The Program | 64 |
| 4.10 | Summary | 67 |
| 5 | (T)ransaction Program | 68 |
| 5.1 | Starting With Reusable Files | 68 |
| 5.2 | A Monthly Run | 73 |
| 5.3 | Examining The Program | 77 |
| 5.4 | Summary | 79 |
| 6 | Report: Income/Expense And Balance Sheet | 80 |
| 6.1 | Running The Program | 81 |
| 6.2 | Examining The Program | 84 |
| 6.3 | Summary | 86 |
| 7 | Invoice Program | 87 |
| 7.1 | Details Of The File System Used | 87 |
| 7.2 | Running The Program | 89 |
| 7.3 | Examining The Program | 98 |
| 7.4 | Customizing The File Structure | 99 |
| 7.5 | Eliminating The Reuse Of Customer Numbers | 99 |
| 7.6 | Customizing Terms of Sale | 100 |
| 7.7 | Deleting The "Get Printer Ready" Message | 100 |
| 7.8 | Making Adjustments For Your Customized Form | 100 |
| 7.9 | Summary | 100 |
| 8 | Mailing Label Program | 102 |
| 8.1 | Running The Program | 102 |
| 8.2 | Examining The Program | 104 |
| 8.3 | Summary | 105 |
| 9 | Checkwriting Program | 106 |
| 9.1 | Running The Program | 106 |
| 9.2 | Using The Supplier's File System | 109 |
| 9.3 | Adding Keyboard Input | 110 |
| 9.4 | Using Transaction File Input | 112 |
| 9.5 | Examining The Program | 112 |
| 9.6 | Summary | 114 |
| 10 | Using TRANSaction To Balance Your Cash Account | 115 |
| 10.1 | Running The Program | 115 |
| 10.2 | Summary | 116 |

## PART III EVALPAC


| | | |
|---|---|---|
| 11 | Statistical and Analysis Menu | 119 |
| 11.1 | **What's In The EVALPAC Package?** | 119 |
| 11.2 | Organization Of This Section | 120 |
| 11.3 | Running Evalpac | 120 |
| 11.4 | Chapters, Routines, And Programs | 121 |
| 11.5 | Examining Evalpac | 121 |
| 12 | Loan Amortization | 123 |
| 12.1 | Running The Program | 123 |
| 12.2 | Theory | 127 |
| 12.3 | Examining The Program | 128 |
| 13 | Break-Even Analysis | 130 |
| 13.1 | Running The Program | 130 |
| 13.2 | Examining The Program | 131 |
| 13.3 | Summary | 132 |
| 14 | Call Balloon Note Value | 133 |
| 14.1 | Running The Program | 133 |
| 14.2 | Theory | 133 |
| 14.3 | Examining The Program | 134 |
| 15 | Declining Balance Depreciation | 135 |
| 15.1 | Running The Program | 135 |
| 15.2 | Theory | 137 |
| 15.3 | Examining The Program | 137 |
| 16 | Future Value Of An Asset | 139 |
| 16.1 | Running The Program | 139 |
| 16.2 | Theory | 139 |
| 16.3 | Examining The Program | 140 |
| 17 | Mortgage Calculations | 141 |
| 17.1 | Running The Program | 141 |
| 17.2 | Theory | 143 |
| 17.3 | Examining The Program | 143 |
| 18 | Present Value of Unequal Cash Flows | 144 |
| 18.1 | Running The Program | 144 |
| 18.2 | Theory | 145 |
| 18.3 | Examining The Program | 145 |
| 19 | Real Estate Property-1981 Depreciation | 146 |
| 19.1 | Running The Program | 146 |
| 19.2 | Theory | 147 |
| 19.3 | Examining The Program | 149 |

| | | |
|---|---|---|
| 20 | Straight Line Depreciation | 150 |
| 20.1 | Running The Program | 150 |
| 20.2 | Theory | 151 |
| 20.3 | Examining The Program | 151 |
| 21 | Value Discounted of a Bond or Mortgage | 152 |
| 21.1 | Running The Program | 152 |
| 21.2 | Theory | 154 |
| 21.3 | Examining The Program | 154 |
| 22 | Depreciation Personal Property 1981-Law With 1982 Revisions | 155 |
| 22.1 | Running The Program | 156 |
| 22.2 | Theory | 156 |
| 22.3 | Examining The Program | 157 |
| 23 | Sum Of The Year's Digits Depreciation | 158 |
| 23.1 | Running The Program | 158 |
| 23.2 | Theory | 159 |
| 23.3 | Examining The Program | 159 |
| 24 | Cash Flow Analysis | 161 |
| 24.1 | Sample Application | 162 |
| 24.2 | Running The Program | 162 |
| 25 | Forecast | 172 |
| 25.1 | Running The Program | 172 |
| 25.2 | Theory | 182 |
| 25.3 | Examining The Program | 184 |
| 25.4 | Program Listings | 187 |

# PREFACE

This book includes a collection of programs written in MBASIC geared for those who are interested in microcomputer business application programs. It uses many of the Apple II-Plus/II-E personal computer's capabilities to perform complex calculations rapidly and present the results on printer and monitor.

The central objective is to provide interested users (small business owners to students) with a broad collection of useful programs to assist in the management of a small business enterprise or, to assist in the learning of microcomputerized accounting. The programs contained herein are written for a specific microcomputer operating system, the CP/M system, that has been tailored for the Apple II/II-E. To use these programs with the Apple II series microcomputer, the user will need an appropriate Z80 peripheral card (and possibly a suitable 80 column video display card) that is available through any certified Apple dealer.

The dialect of BASIC used in the programs is Microsoft's BASIC-80, version 5.0. This version of MBASIC has been modified slightly to allow use of Applesoft BASIC's commands such as VTAB, HTAB and HOME. However, the Apple version of MBASIC is a more powerful version of BASIC than the current Applesoft BASIC and contains some commands not supported by Applesoft BASIC. In addition, these programs are made to run under control of the popular CP/M operating system as designed by Digital Research. CP/M (which stands for Control Program/Microprocessors) is used on most of the popular 8-bit microcomputers. To date, CP/M is the most popular operating system in the industry and should continue to represent the industry standard for 8-bit micros for several years.

Programs in this collection may be divided into 2 groups. The first is designed to produce balance sheets and profit and loss statements for a small- to medium-sized business. Included are programs that write checks and invoices, and produce the general ledger results. Such programs would allow for a small business manager to start computerizing his or her business.

The second collection of programs is for business analysis and evaluation. Included is a program that performs many different financial analysis routines. The programs can produce an amortization table and depreciation tables of all types, and perform complete analysis of real estate investments. Some of the statistical programs are designed to calculate the values of regression lines and plot the lines with confidence and prediction limits. These programs, in one form or another, are generally needed by most business analysts.

A word of warning to the interested user is in order. This book is not intended for the first-time microcomputer user. It is presumed that the reader has had some prior experience with microcomputers and has accomplished some individual programming in BASIC. However, one's experience with micros and with programming need not be too "extensive." Simply stated, a few months or so of experience with the Apple system and with Applesoft BASIC will suffice for most users. In addition, some prior accounting background would

be helpful. However, one of our objectives is to assist in the understanding of the day-to-day applications of accounting, and the book could well serve as an instructional aid in learning fundamental accounting procedures.

There are a number of people who must be acknowledged for their direct and indirect contribution to this book. First, special recognition must go to our wives, Carol and Dianne, who suffered through the many hours of programming, debugging, manuscript writing, and editing. Without their patience, encouragement, and understanding, very little of this book would have been possible.

*Steven M. Zimmerman, Ph.D.*
*Mobile, Alabama*

*Leo M. Conrad*
*Mobile, Alabama*

*Donald R. Smith, Ph.D., P.E.*
*College Station, Texas*

Special thanks to Robert and David for their help

# LIMITS OF LIABILITY AND DISCLAIMER OF WARRANTY

The authors and publisher of this book have used their best efforts in preparing this book and the programs contained in it. These efforts include the development, research, and testing of the theories and programs to determine their effectiveness. The authors and publisher make no warranty of any kind, expressed or implied, with regard to these programs or the documentation contained in this book. The authors and publisher shall not be liable in any event for incidental or consequential damages in connection with, or arising out of, the furnishing, performance or use of these programs.

# TRADEMARKS OF MATERIAL MENTIONED IN THIS TEXT

APPLE II, IIE: APPLE COMPUTER CORPORATION
6502 MOS CHIP: SYNERTEK
CP/M+ CARD: ADVANCED LOGIC SYSTEMS (ALS)
Z80 CARD: MICROSOFT
MBASIC: MICROSOFT
CP/M: DIGITAL RESEARCH
Z80: ZILOG CORPORATION
MX-80, MX-100: EPSON AMERICA INC.

# I: The Apple II And Its Systems

# 1. The Apple II Microcomputer System with CP/M

The Apple II microcomputer is a home or business computer capable of growing into a powerful professional/business machine with enough capacity to handle the needs of most small- and medium-sized businesses. Since the inception of the Apple II microcomputer in the late 1970's, it has become vastly popular in the home, business and educational fields. Literally hundreds of games and application programs have been developed for the Apple II system.

The Apple II is marketed in modules, which means that the startup configuration may be kept within the budget of the home user. The basic unit consists of the keyboard and a "mother-board" ready to receive additions. A basic configuration for the Apple II would most likely consist of the keyboard/mother-board unit, one disk drive and a video monitor. Such a system will retail for under $2000.00.

In January of 1983, Apple Computer introduced a new version of the already popular Apple II Plus system. This new system is termed the Apple IIe (the "e" stands for "enhanced"). The IIe system consists of a keyboard/mother-board unit, one disk drive and an Apple III green screen monitor. The retail price for this "starter system" was established at $1995.00. In effect, the IIe replaced the II Plus system. However, thousands of II Plus systems were purchased by users, and Apple designed the IIe to be able to run nearly all of the software that was developed for the II Plus.

The real flexibility of the Apple II system lies in the fact that inside the keyboard unit there are 8 expansion slots capable of receiving printed circuit cards called "peripheral cards." These cards can perform a variety of functions such as printer communications, additional memory expansion, serial communications, and a host of other tasks. By the judicious use of the expansion slots, the user of the Apple II system can custom design the system to suit a variety of individual needs.

## 1.1 What Does It Look Like?

The Apple II system consists of a main keyboard/mother-board unit housed in an attractive durable case. The central processor of the Apple II is Synertek's MOS Technology 6502 8-bit microcomputer chip. As designed for the Apple II, the chip runs at a rate of 1,023,000 machine cycles per second and is capable of addressing 65,536 "bytes" of information. In computer jargon, the term "byte" refers to an addressable memory location capable of storing and retrieving a single character (digit, letter, symbol, space).

An 8-bit microcomputer utilizes a pattern of eight "bits" to represent one byte. (The term "bit" stands for BInary digiT.) Thus, characters stored within the memory of an 8-bit microcomputer are comprised of patterns of eight bits, where a bit assumes a numerical value of either zero or one. In addition to the 6502 chip, other circuits are in place to support the required functions of

memory storage, keyboard communications, disk input/output, game I/O controls, and other technical requirements. A power supply unit located on the left side of the interior is used to convert the A/C input voltage and current to reduced voltage levels required by the microcomputer circuitry. Figure I-1 illustrates a typical Apple II Plus system configuration. Figure I-2 illustrates the Apple IIe system.

**Fig I-1: Apple II Plus System**

Within the keyboard unit, in addition to the eight expansion slots and the 6502 chip, is either 48k or 64k of Random Access Memory (RAM) and 12k bytes of Read Only Memory (ROM). RAM memory is used for the temporary storage of data and program statements and is termed volatile memory. The term "volatile" means that when the power is turned off all of the data and instructions stored in RAM are lost. RAM is reactivated and ready to receive new data or instructions when the power is turned on again. ROM memory, on the other hand, is termed "non-volatile" memory because the instructions are burned into the chips during the manufacturing process and cannot be altered unless it is erased and new instructions burned in. Thus, the special instructions present in ROM memory remain intact regardless of the power status and are available each time the computer is turned on. The contents of ROM memory can only be read by the system and not written to or altered by the normal process of operations. Hence the term "Read Only Memory."

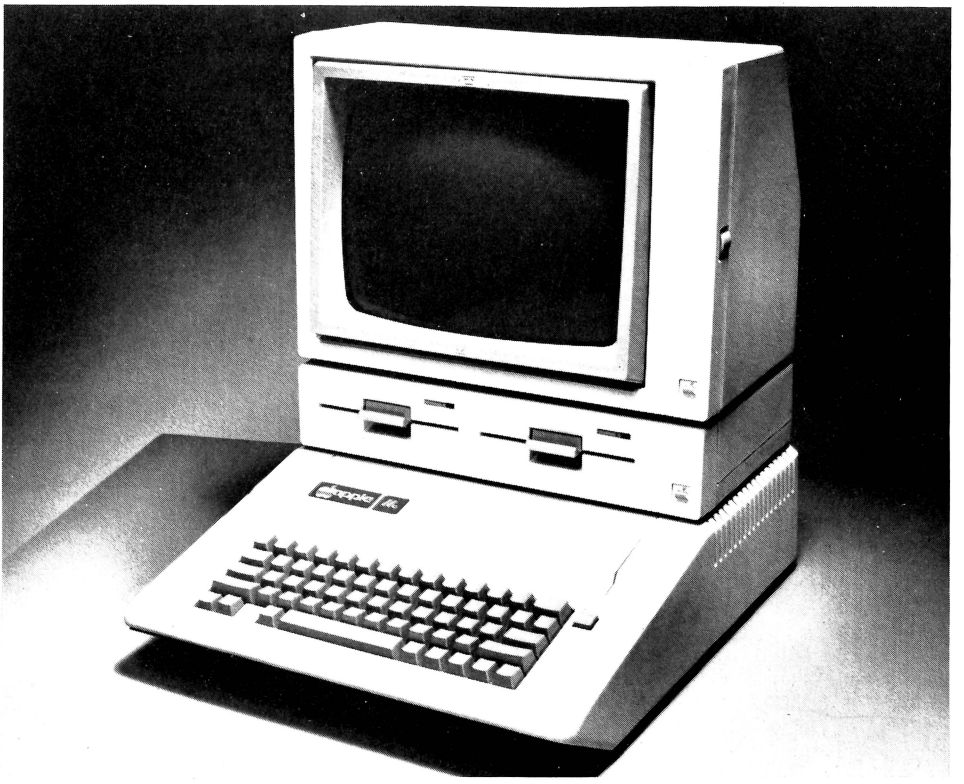


**Fig I-2: Apple IIe System**

The ROM memory includes the system monitor, utility subroutines, programmer's aids, and two dialects of the BASIC language. The Apple II system is capable of being programmed in either Integer BASIC or a more powerful level of BASIC called Applesoft BASIC. A bootstrap program used to start the disk system is also contained within ROM. This program is called Autostart ROM and is used to load the operating system stored on the booting disk.

## 1.2 What Is It?

The Apple II system is a powerful microcomputer built around the 6502 microprocessor chip, which provides more potential and capabilities than most people can imagine for personal computing. Extremely versatile, the Apple II is useful for your children's studies, be they first graders, mechanical engineers, business administration majors, landscape gardening majors, or medical students. For word processing, it has several of the largest electronic worksheets now available and an accounting system for small- or medium-sized businesses. It can also be used as a control device for your home or even specialized laboratory equipment.

From the multitude of peripherals being offered by Apple and other

manufacturers, you have a choice of scientific instrumentation, plotters, hard disks, light pens, digitizers, graphic tablets, home protection devices, voice synthesizers, and much more. Figure I-3 illustrates a typical Apple system with, some of the more popular peripheral devices that can be attached to the system.

**Fig. I-3: Apple II with accessories**

## 1.3 Software To Run The Apple II System

A computer consists of electronic hardware you can see and touch, system programs that run the machine, and application programs to perform specific tasks. Programs are sets of instructions you can purchase on magnetic tape or diskettes. Programs are software that instruct the computer to produce the product desired.

Software packages, whether from Apple or some other software manufacturer, are available "off the shelf" to allow you to do most anything with your Apple computer. This software provides you with endless opportunities to produce useful results and enjoy this fantastic computer.

In Part II of this book you will find a number of programs designed to operate under control of the Apple II provided certain hardware/software additions are installed. The programs in this book are designed to run under control of an operating system called Apple CP/M, which is not the normal operating system that comes with the Apple computer. The on-board operating system is termed Apple-DOS, which stands for Apple Disk Operating System. Chapter II will define and explain the differences and document why CP/M (which stands for Control Program/Microcomputers) was selected. Chapter II will also explain the functions of an operating system. The accounting programs described in subsequent chapters will assist the user in the day-to-day operation of a small business enterprise. Following the chapters relating to the accounting system are chapters relating the evaluation of certain quantitative business problems. Included in this set are programs relating to capital recovery, amortization, and real estate investment evaluation.

Commercial software to do almost anything you can think of is available for your Apple. More programs are being produced constantly. The most commonly purchased programs include word processors, electronic worksheets, data management systems, and accounting systems. With the Apple you have your choice of many different programs in all groups. Your biggest problem is selecting the one best suited to your needs. Your Apple dealer, other Apple users, and microcomputer oriented magazines are good sources of information about these programs.

## 1.4 Examining Some Add-Ons

What is it you need? Analog to digital/digital to analog converters to run your lab or business? BSR 10 appliance or light switch controllers to make your house look lived in while you're away? IEEE-488 Interface unit to run laboratory equipment among other devices? Multifunction cards offering two, three, or four port positions for communications or printing functions; Winchester drive hard disks from 5 to 20 Mg (Megabyte = 1000 KB) formatted; real time clock and calendar with rechargeable nicad battery; voice synthesizer and speech synthesis; shared device controllers; stepper motor controllers; bus extender cards; video bit mapped graphics generators; plotters; digitizers; local area networks (LAN)? The list has hardly begun.

Applications or specific uses you want for your Apple II computer will determine what special attachments or peripherals you may need. It is how you wish to use your computer that leads manufacturers or inventors to come up with answers to your needs. With all these add-ons available, proper planning for the growth and expansion of your Apple II can be rewarding to your pocketbook and your computing experience. If you were fortunate enough to purchase a "full blown" system when you got your computer, you probably began with 48 KB of RAM memory, 2 disk drives, green screen monitor parallel printer interface, and a dot matrix printer such as the Epson MX-80 or MX-100. This configuration is one of the most common sold. This book assumes you have a computer configured in this manner or with similar capabilities.

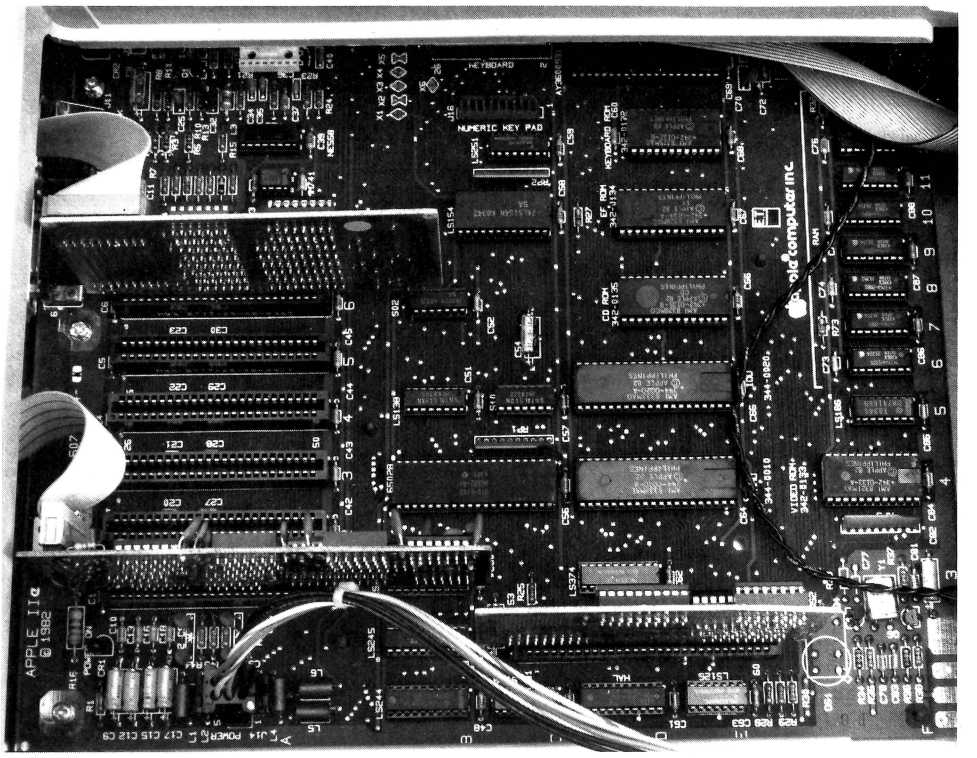



**Fig. I.4. Selected Devices and Interface Cards**

To those of you who started out on a lower budget, your first purchase very likely will be the first disk drive followed by a second or a printer. At that point, discussion of planned additions for your system will begin. Depending on your main interest in computing (that is, word processing, accounting, graphics, scientific analysis, engineering, studying, programming, or even "hobbyist"), try to plan your next requirement for add-on and schedule your budget that way.

For word processing, your major consideration is the purchase of a software system to support word processing, the addition of a letter quality printer, a software package to support spelling checking of your documents, a cut sheet feeder for the printer, and possibly some of the new smooth-edged, perforated, tractor-fed paper. Letter quality printers with the daisy wheel, thimble, or ball typing heads have been relatively expensive in the past. Their prices, like many others in the computer field, have been coming down. You must shop carefully in this rapidly changing market to obtain the best buy for your needs. Several daisy wheel printers are now available for under $1,000, but do not let price alone influence your purchase. Make sure the printer will do what you need it to do and that it meets all your needs or expectations.

Cut sheet feeders often cost as much or more than the printer itself. With finely perforated bond paper now available, you may consider using the pre-

printed letterhead pin-feed or fanfold tractor paper. It is expensive, but, when weighed against the cost of a sheet feeder, it may be a better buy. Meanwhile, a growing number of good dictionaries or spelling checkers are available for your Apple system. Their sophistication also varies, from a simple root word checker to a writing style analysis. Look them over carefully. Ask your dealer questions, and if he doesn't know the answers, ask him to find out for you. After all, when you buy from an authorized dealer you expect support.

## 1.5 Setting Up Your System

Computers are made to be used. You should take extra care in locating your system. If you expect the computer to produce accurate results, the operator should not be interrupted constantly as he or she inputs data and runs the system.

The physical arrangement is also important. There are many specialized computer furniture manufacturers that make equipment specifically for your computer. If you can afford such equipment, it does a great job of matching the operator with the equipment. If you are using standard desks, make sure the working area is large enough to accommodate the equipment. Most computers will require a 30-inch desk depth and should be 48 to 60 inches wide.

Your printer must have space for the storage of new and completed paper. It is very important to have the paper feed freely into and out of the printer, or problems will develop.

## 1.6 Getting Started

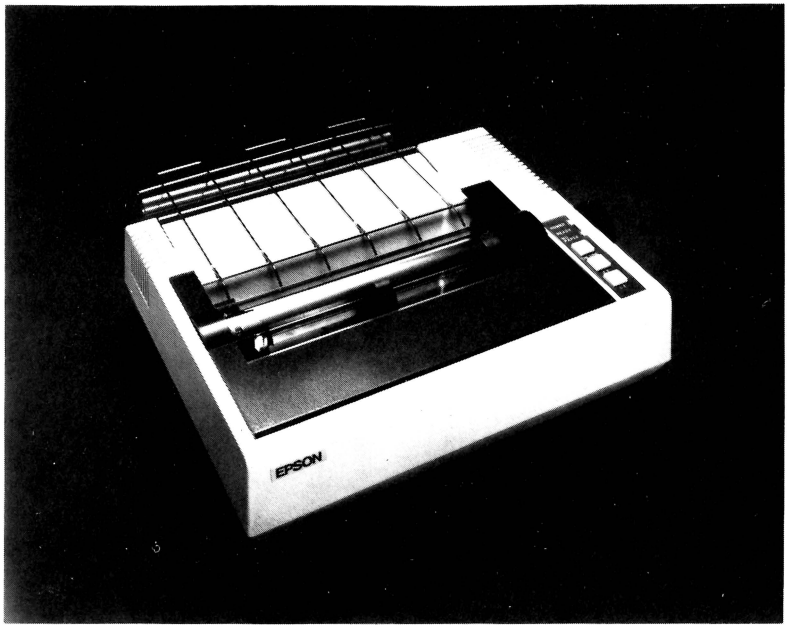
We recommend that you obtain a power strip with some type of power surge protection, plug your computer and all its peripherals into it, and use the switch on the power strip to turn all your equipment on and off at the same time. Your Apple II has some surge protection built in, but the added protection helps, and the switch lets you control the entire system from a single location.

Unlike many other computers, your Apple II is designed to be turned on with a system diskette in drive 1. Apple numbers its disk drives 1,2 but, in Chapter II, a different drive notation will be used to conform to the CP/M drive notation. (CP/M denotes the disk drives as Drive A:, B:.) When you purchase the required add-ons that will be described shortly, be sure to place decals on your Apple Disks as follows:

Drive 1 will become A:

Drive 2 will become B:

After installing a diskette in drive A and pushing the power switch, there is a slight delay. The length of the delay is a function of the Autostart ROM program.



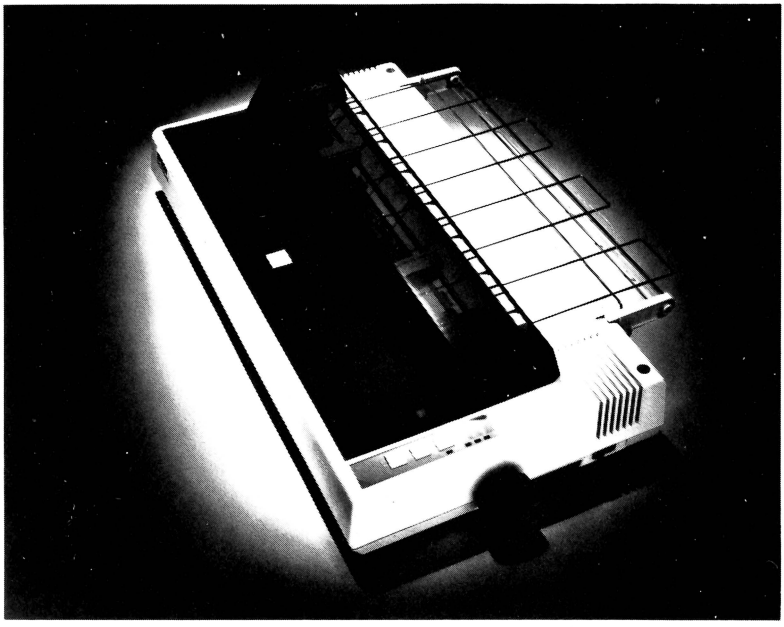



Fig. I.5. MX 80 AND MX 100 Series Epson Printers

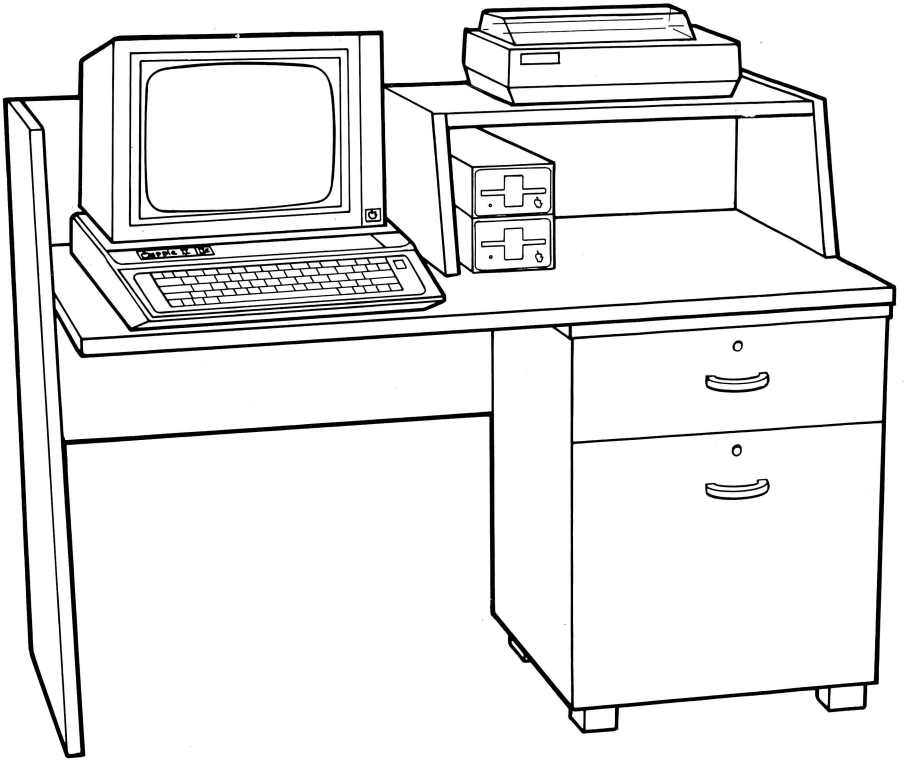


Fig. 1.6. Apple II on a special design desk

## 1.7 Printer Graphics

It is difficult to guess what printer an individual has attached to his or her computer. Two printers are popular with owners of Apple II computers: the Apple printer and the Epson MX-80 or 100 printer manufactured by Epson Corporation. These printers may be equipped with Graphtrax, allowing them to print graphics characters. The graphics programs are designed to run using Microsoft's M-BASIC under control of CP/M with a minimum of 64 KB of memory.

## 1.8 Copying Programs

You may obtain the programs for this book either by typing them directly into the computer yourself or by purchasing a diskette with the programs on it. The diskette with all programs contained in this book is available from the publisher at a reasonable cost. It is assumed that those who are going to copy the programs have some programming knowledge or are in the process of learning programming. If you are planning to copy the programs yourself, you will find the section below about copying disks of some value.

The diskette on which the programs have been placed is a single-sided diskette designed to operate in the 140Kb disk drive of the Apple II. There is no operating system, nor is there a copy of MBASIC on the diskette. You have the option of copying the operating system to the diskette or copying the programs from the diskette to a blank, formatted diskette. This assumes that you have an appropriate Z80 peripheral card , 80 column card for 80 column video display, and a CP/M systems master diskette containing MBASIC.

The next section will present the step-by-step process for copying an Apple CP/M formatted diskette. If you have had little experience with Apple CP/M, you may wish to stop here and read Chapter II. Additional references pertaining to CP/M may be found at the end of Chapter II.

## 1.9 For Dual Drive Systems

If you wish to copy the CP/M operating system and MBASIC to the program diskette, we refer you to your CP/M software manual. The procedure outlined below is for copying the programs to a formatted diskette and assumes that you have two disk drives. However, before reading further, you may elect to review Chapter II in order to understand the Apple II CP/M procedures. If you are already familiar with Apple CP/M, you should experience no difficulty in understanding the copying process.

Step 1. Place a CP/M master diskette in drive A and power-up the system. When you receive the CP/M prompt A>, you are ready to enter a CP/M command. Type in the command FORMAT <CR>. Throughout this book the term "<CR>" means to depress the RETURN key on the keyboard. CP/M responds with with a three-line promo telling you that the 16-sector Apple CP/M formatting program is now loaded. The prompt below this promo will say:

```
Format disk in which drive?
```

Your response should be:

```
B:<CR>
```

CP/M will respond with:

```
Insert disk to be formatted in drive B:
Press RETURN to begin.
```

Before pressing the <CR> key, be sure you have placed a blank diskette in Drive B: and closed the drive latch. At this point, go ahead and press <CR>. The in-use red light on Drive B: will come on, and you can hear the read/write head indexing inside of the drive. The formatting procedure will take approximately 20 seconds. When the formatting process is concluded, Apple CP/M will respond with:

```
Format disk in which drive?
```

This means that the formatting process is finished and the system is ready to

format another diskette. If you have other disks to format, simply remove the previously formatted disk from drive B: and insert another diskette in its place. Respond with B: and press <CR>. The formatting process will repeat. If you are finished formatting, you are now ready for step 2. If you make a mistake or wish to exit, simply press "ctrl-C" (depress the CTRL key and the key marked "C" at the same time and repeat the process).

Step 2: At this point you have the prompt A>. Enter the following command from the keyboard:

```
COPY B:=A:/S <CR>
```

The Apple CP/M prompt will respond:

```
Insert MASTER disk into drive A:
Insert SLAVE disk into drive B:
Press RETURN to begin
```

At this point, press <CR>. This activates the copy routine which will place a copy of the CP/M System on the formatted diskette. This is the purpose of the /S option in the COPY command given above. A copy of the CP/M operating system is placed on the diskette. This process will take only two or three seconds. When completed, CP/M will respond:

```
Copying...
COPY complete
Do you wish to make another COPY?
```

If you respond with a "YES", the same prompts are repeated, which will allow the copying of the CP/M system on other diskettes placed in Drive B:. If your response is "NO" or simply "N", CP/M responds with:

```
Insert CP/M System disk into Drive A:
Hit RETURN
```

CP/M will respond with the normal prompt:

```
A>
```

At this point, you have a CP/M formatted diskette with CP/M copied onto the appropriate tracks of the diskette. However, you have nothing else! A file called MBASIC and the various programs that make up the accounting and business analysis application programs described in this book are needed. In order to copy MBASIC to the diskette, the following procedure is required:

1. Place a copy of the CP/M master diskette in Drive A: and power-up the system.
2. After receiving the CP/M trademark logo and the CP/M prompt A>, enter the following from the keyboard:

```
DIR <CR>
```

The DIR command provides you with a directory of the Master Diskette. The screen will display 19 CP/M files that come with CP/M diskettes. This assumes that you have purchased the Microsoft Softcard and the asso-

ciated CP/M software, At the bottom of the file listing the CP/M, prompt A> will appear. Type:

```
PIP B:=A:MBASIC.COM <CR>
```

PIP is a CP/M file copying utility and stands for Peripheral Interchange Program.

3. To validate the file transfer enter the command:

```
DIR B: <CR>
```

CP/M will respond with the screen display:

```
B: MBASIC COM
```

which validates the fact that MBASIC was copied over to Drive B:. At this point, you have MBASIC resident on the diskette in B:.

Step 4: When the transfer of MBASIC to the diskette in Drive B: is complete, the next task is to copy the program diskette containing the accounting and business analysis programs over to the Drive B: diskette. In order to accomplish this task, place the CP/M master diskette in Drive A: and enter from the keyboard the following CP/M command:

```
COPY B:=A:<CR>
```

CP/M will respond with:

```
Insert MASTER Diskette into drive A:
Insert SLAVE Diskette into drive B:
Press RETURN to begin
```

At this point DO NOT press <CR> yet. Remove the System Master from Drive A and replace it with the program disk containing the accounting and business analysis programs associated with this book. When this diskette is properly loaded and the disk drive door latch is closed, then press <CR>. CP/M will respond with the statement:

```
Copying ...
```

After approximately 10 to 15 seconds, CP/M will respond with:

```
COPY Completed
Do you wish to make another copy?
```

If you respond with Y, CP/M will return to the instruction prompt shown above. We suggest that you format at least three diskettes as described is steps 1-3 and make back-up copies. If you respond with a "N", CP/M will prompt you with:

```
Insert CP/M System into Drive A:
Hit RETURN
```

Before you hit return, replace the CP/M system master into Drive A: and close the latch. Then hit the <CR> key. CP/M will access Drive A: and return you to the normal CP/M system prompt A>.

Step 5. File the program diskette in drive A in a safe place. Make copies of the diskette in drive B as needed.

You may want to make three copies of the diskette in drive B and then erase all programs except the accounting set from one diskette and all programs except the analysis set from a second diskette. The programs and the sets into which they fall are as follows:
The accounting general ledger set is:

```
1. MENU.BAS
2. CHECK.BAS
3. INVOICE.BAS
4. CPA.BAS
5. CHART.BAS
6. TRANS.BAS
```

The business evaluation set is:

```
1. MENU1.BAS
2. EVALPAC.BAS
3. CASH.BAS
4. FORECAST.BAS
```

## 1.10 For Single Disk Drive Systems

Copying programs with a single disk drive (Drive A) can be a tedious task. In general, copying an Apple CP/M disk represents a two-step process. The following procedure is used to perform a single disk copy of any Apple CP/M disk:

Step 1. Load an Apple CP/M system master into Drive A and power-up the system. When the CP/M prompt appears enter the following command:

```
FORMAT A:<CR>
```

Within several seconds, the Apple will respond with:

```
APPLE II CP/M
16 SECTOR DISK FORMATTER
(C) COPYRIGHT MICROSOFT 1980
INSERT DISK TO BE FORMATTED INTO DRIVE A:
```

At this point, remove the CP/M master from the drive, insert a blank diskette, and close the drive latch. Press <CR> only after you are sure the disk that you have inserted is blank or is a diskette that you intend to erase. After pressing <CR>, the disk formatting routine will take approximately 30 seconds. The disk drive will remain on during the entire process. When the formatting process is completed, the disk drive will stop and the Apple will display:

```
FORMAT COMPLETED
INSERT SYSTEM DISK AND PRESS RETURN.
```

When the drive light goes out, remove the newly formatted diskette from the drive, re-insert the CP/M master diskette, and close the drive latch. Then, press <CR> to reactivate CP/M. After a second or two, the A> prompt will appear, which signals the fact that you have returned to CP/M.

Step 2: At this point you must copy the Apple CP/M operating system

(briefly described in Chapter II) onto the formatted diskette. This is accomplished as follows:

a. Insert an Apple CP/M master diskette into Drive A: and power-up the system. Enter the command:

```
COPY A:=A:/S<CR>
```

The "/S" means that you want to copy only the CP/M operating system onto the newly formatted diskette. Apple CP/M will respond with:

```
INSERT MASTER DISK PRESS RETURN.
```

Since the disk in Drive A is an Apple CP/M master, just leave the current disk in Drive A and press <CR>. The disk will whirr for a few seconds and the Apple will display:

```
INSERT SLAVE DISK PRESS RETURN.
```

At this point, remove the CP/M master disk and replace it with the newly formatted disk, close the drive latch, and press <CR>. After a few seconds, the Apple will respond with the prompt:

```
INSERT CP/M SYSTEM DISK INTO DRIVE A PRESS RETURN.
```

Remove the current disk and replace it with the Apple CP/M master diskette, close the drive latch, and press <CR>. When the CP/M prompt A> appears, you have returned to CP/M and the system is awaiting further CP/M system commands.

Step 3: Now you are ready to copy from any Apple CP/M disk. Enter the following from the keyboard:

```
COPY A:=A:<CR>
```

Within a few seconds, Apple CP/M will respond with:

```
APPLE II CP/M
16 SECTOR DISK COPY PROGRAM
(C) MICROSOFT 1980
INSERT MASTER DISK PRESS RETURN.
```

The term "master disk" means any Apple CP/M diskette that contains data and/or programs. Assuming you desire to copy the diskette containing the programs for this book, insert the diskette into the drive (Drive A:), close the drive latch and press <CR>. The drive will be activated for a few seconds, then the Apple will respond with:

```
INSERT SLAVE DISK PRESS RETURN.
```

Remove the diskette containing the desired program(s) from the drive and replace it with the freshly formatted diskette. Close the drive latch and press <CR>. After a few seconds, the Apple will respond with:

```
INSERT MASTER DISK
PRESS RETURN.
```

Before pressing <CR>, remove the diskette from the drive (this diskette is termed the "slave" or "backup" disk) and re-insert the master diskette, close the drive latch, and hit <CR>.

The computer will ask you to re-insert the slave diskette. THIS PROCESS WILL BE REPEATED THREE TIMES.

After you have inserted the backup diskette into Drive A: for the last time, the Apple will display:

```
COPY COMPLETE
DO YOU WISH TO MAKE ANOTHER COPY? (Y/N)
```

If you press "Y" for yes, the above procedure will be completed. If you enter a "N" for no, the command

```
PRESS RETURN
```

will be displayed. At this point, the disk in Drive A is an exact copy of the master diskette you wanted to copy. Normally, you should plan to make at least three backup copies of all program and data disks and store the original plus one copy in a safe place, well away from temperature extremes and magnetic fields. At this point, your backup diskette contains the CP/M operating system and the programs described in this book.

Step 4: The final step in preparing your program diskette is to copy the MBASIC language file over from the Microsoft MBASIC disk. If you purchased the Microsoft Apple CP/M system , your CP/M system's master diskette contains a file called "MBASIC.COM". It is imperative that MBASIC be copied over to the diskette you just configured in the preceding steps.

In order to perform the transfer of one or more files from one diskette to another, Apple CP/M utilizes a utility called PIP.COM. PIP stands for "Peripheral Interchange Program" and is mainly used to transfer files from one diskette to another. If you are not experienced with CP/M, you will soon discover that PIP is one of the most utilized CP/M utilities. Since it is not the objective of this text to provide in-depth detail of CP/M, only the essential aspects of PIP will be covered. There are several outstanding textbooks describing CP/M that are readily available from your Apple dealer and most bookstores that carry microcomputer subject matter. A list of the more popular CP/M references is provided at the end of Chapter II.

To transfer MBASIC from your CP/M master diskette to your accounting diskette, the following procedure is utilized:

a. Place a CP/M systems master diskette in Drive A, close the latch, and power-up the system. If the system is already on the diskette, place a Systems Master diskette in Drive A and press CTRL-C. The system will respond with the normal CP/M prompt A>.
b. Place the CP/M formatted accounting diskette in Drive B: and close the drive latch.
c. Type the following CP/M command from the keyboard:

```
PIP B:=A: MBASIC.COM<CR>
```

The PIP program will perform the transfer of the file called MBASIC.COM located on Drive A to the diskette located in Drive B:. The transfer will take 10 to 15 seconds. When the transfer is completed, the monitor will display the normal CP/M prompt A>.

d. To validate the file transfer, we will examine the disk directory of the accounting diskette by entering the following CP/M command:

```
DIR B:<CR>
```

For the accounting package, the screen display will show the following:

```
B: MBASIC COM: TRANS BAS: CPA    BAS: INVOICE BAS
B: MENU    BAS: MAIL  BAS: CHART BAS: CHECK   BAS
```

The above files are required to execute the accounting package in Apple CP/M.

The same disk formatting and file transfer for the Business Evaluation package master diskette is also required. Therefore, repeat the CP/M formatting and PIP procedure previously described to obtain a working copy of the EVALPAK system. When correctly copied, the EVALPAK Apple CP/M diskette should contain the following files:

```
1. MBASIC COM
2. MENU1 BAS
3. EVALPAC BAS
4. CASH BAS
5. FORECAST BAS
```

## 1.11 Summary

To summarize, we suggest that you make backup copies of your CP/M master diskette, accounting package diskette, and the EVALPAK system. It is a good idea to make three copies of all disks. Store the original diskette and one backup copy in a secure location. The other two diskettes can be used as a working copy and a backup working copy.

Thoroughly read the CP/M manuals that come with your particular Apple CP/M system and become familiar with Apple CP/M. The programs we will describe have been designed for the Microsoft Z- 80 Soft Card. When you purchase this system, you will receive the interface card (which goes in slot 4 of the Apple computer) and two Apple CP/M diskettes. One of the diskettes is configured for the Apple 13 sector disk format and the other is configured for the Apple 16 sector format. It is most likely that you will use the 16 sector master diskette. In addition, Microsoft will provide two reference manuals that describe the features of Apple CP/M and Microsoft BASIC.

It should be pointed out that other brands of Z80 cards now exist for the Apple II/IIe system. An authorized Apple dealer can assist you in your purchase so long as the card will support CP/M 2.0 and MBASIC Version 5.0. In addition, the purchased system must format diskettes in the same manner as Microsoft's Z80 Softcard.

The next chapter will review some of the main features of Apple CP/M. It will include only the highlights of Apple CP/M so the interested reader will need to refer to more in-depth textbooks and reference materials relating to CP/M.

## GENERAL REFERENCES

*YOUR SMALL BUSINESS COMPUTER* by Donald R. Shaw, Van Norstrand Reinhold Co., New York, N.Y., 1981.

*APPLE II USER'S GUIDE* by Lon Poole, Martin McNiff and Steven Cook, Osborne/McGraw-Hill, Berkeley, CA.,1981.

*BASIC PROGRAMMING SELF-TAUGHT* by Seymour C. Hirsch, Reston Publishing Company,Inc., Reston, VA., 1980.

*BASIC-80 and CP/M* by Jack Jay Purdum, Macmillan Publishing Co., Inc., New York, N.Y., 1983.

*MICROSOFT BASIC* by Ken Knecht, Dilithium Press, Beaverton, OR., 1979.

*COMPUTER STATION'S PROGRAMMER'S HANDBOOK to the APPLE II*, Computer Station, 11610 Page Service Drive, St. Louis, MO., 63141, (314) 432-7019, 1981.

*APPLE II REFERENCE MANUAL*, APPLE Product No. A2l0001A, Apple Computer, Inc., 10260 Bandley Drive, Cupertino, CA 95014, 1981.

*MICROSOFT SOFTCARD USER'S MANUALS (VOL. 1 & 2)*, MICROSOFT CONSUMER PRODUCTS, 400 108TH AVE. NE, SUITE 200, BELLEVUE, WA. 98004, 1980.

# 2. APPLE CP/M Operating System

There are many different operating systems available for your APPLE II computer. One of the systems sold by APPLE is called APPLE DOS (short for Apple Disk Operating System) and is an excellent system for most Apple applications. Some of the other systems have specialized abilities that make your Apple computer easier to use for specific applications. However, the most popular operating system add-on is the Apple CP/M System promoted by Microsoft. The programs presented here assume that the user has purchased the Microsoft Z-80 Softcard or an equivalent add-on Z-80 card and has a computer with 80-column capability and two disk drives with a line printer.

At the writing of this book, Advanced Logic Systems (ALS) of Sunnyvale, California, has introduced a new Z-80 card for the Apple IIe called the CP/M+ Card. Also, ALS included the latest revision of CP/M called "CP/M 3.0." CP/M 3.0 represents an enhanced version of the popular CP/M 2.2 system and possesses faster disk I/O routines. This chapter will concentrate, however, on the Apple CP/M 2.2 that is commonly found on the Apple II Plus systems. All programs contained in this book should run under the control of CP/M 3.0 and MBASIC 5.0 or higher with little or no modification. Before we examine how an operating system works, let's look at the important tasks that any microcomputer operating system must perform:

1. Controlling communication with the console.
2. Controlling additional peripherals (additional disk drives,printers, hard disks, etc.)
3. Controlling the file storage media (disk drives) and how data and programs are stored on the diskettes.

Simply stated, the operating system manages the system via a collection of pre-written software programs. It is the operating system that makes the entire system work as an integrated unit instead of a collection of unrelated parts.

All microcomputers are organized logically into several interrelated parts that must communicate in a coordinated manner. These include the central processing unit (CPU), the main system memory (RAM) and the associated peripherals such as the keyboard console, printer, telephone modem, plotter, and the disk drives. The disk operating system (DOS) is comprised of a series of software programs and user interface commands that control and coordinate the operation of the various components. CP/M represents one of the most popular operating systems for 8-bit microcomputers.

Since CP/M was developed for the 8080, 8085, and Z80 microcomputer chips, one could ask, "Why CP/M for the 6502-based Apple II/II-E?" Simply stated, by the addition of the Z80 Softcard, the power and versatility of the Apple is more than doubled and literally hundreds more applications programs can be executed on the Apple with the CP/M add-on. With the Z80 card and

CP/M, the Apple system is virtually two computers in one (although only one type of DOS can be in operation at one time).

In general, an 8-bit microcomputer CPU can directly address 65,536 bytes of memory. Each byte of memory is similar to a small mailbox capable of holding 8 bits of information and possesses an individual data address. Thus, each byte or character of RAM is assigned an address from 0 to 65,535. The CP/M operating system divides this memory into five basic regions. The five regions, beginning with the lowest memory address, are:

- THE SYSTEM PARAMETER AREA - an area of RAM that contains information relating to the current disk number, peripheral assignments, key addresses for other important parts of CP/M, restart locations, and some buffer storage.
- THE TRANSIENT PROGRAM AREA (TPA) - represents the main working area of RAM where programs and data are stored.
- THE CONSOLE COMMAND PROCESSOR (CCP) - a small area of RAM that saves the operating system built-in commands DIR, REN, TYPE, SAVE, ERA, and USER.
- THE BASIC DISK OPERATING SYSTEM (BDOS) - an area of RAM that stores the general programs for managing and operating all peripherals.
- THE BASIC INPUT/OUTPUT SYSTEM (BIOS) - an area of RAM that stores the customized programs that operate the peripherals.
- Taken together, the BDOS and BIOS segments of RAM are called the FULL DISK OPERATING SYSTEM (FDOS).

To fully appreciate the capability of any microcomputer, the user must, over time, become proficient in the use of the DOS command structure. The version of CP/M designed for the Apple by Digital Research is not exactly the same CP/M found on other 8-bit micros. The reason for this is that the Apple does not use a Z80-based CPU. However, this is overcome by the purchase of a Z80 add-on card giving the Apple access to a Z80 CPU and hence to CP/M as an alternative operating system.

Before we examine some of the relevant features of Apple CP/M, we will review some important aspects of your computer's hardware.

## 2.1 Your Computer's Hardware

Your computer's hardware is a collection of pieces of equipment. Included in this collection are:

1. The video display tube (VDT), also called a monitor or cathode ray tube (CRT).
2. The keyboard.
3. Read Only Memory chips called ROM.
4. Random Access Memory chips called RAM.

5. Disk drives, hard disks, or cassette recorders to store programs and data.
6. A metal or hard plastic case to store the disk drives and memory chips.
7. A central processing chip to run all the other parts of the collection. In your Apple II, this chip is a 6502 processor.
8. Additional electronic boards such as an asynchronous communication board, disk drive controller, display driver interface, printer interface, etc.

The 6502 chip we spoke of is the center of your system. It is called the "central processing unit." This chip controls the many capabilities of your Apple II computer. By itself, however, the 6502 cannot receive instructions from you and then direct the actions of the other parts of the computer system. It needs help in the form of an operating system which ties the parts of the system together. The operating system your Apple II needs must be very powerful, because your computer has many capabilities and needs.

## 2.2 Getting Your Operating System Started

The first program needed by all disk operating systems is a "boot" program that helps the system start itself. This program works with part of the ROM to start the process of loading what is on the disk into the memory of the computer. Once the disk operating system is loaded, it takes over operation of the computer.

The program loaded into your computer by the boot program will vary, depending on the particular operating system being used. In its normal mode, the Apple boots into its own operating system (called Apple DOS) which remains in force unless rebooted. However, with the installation of a Z-80 card (normally in slot 3, 4 or 7), the Apple has the capability to function as a Z-80 microcomputer. An important question is, "How does the Apple know which operating system to come up in?" The answer to this question is centered on the diskette used to boot the system. If the diskette is an Apple DOS 3.2 or 3.3 version, then the Apple will come up in its standard operating system. If the diskette is formatted in CP/M (as described in the latter part of the Chapter 1,) then the Apple system will boot into CP/M and utilize the power of the Z-80 chip on the Softcard. Basically, the Z-80 card is a printed circuit card that can be located in either slot 3, 4 or 7 of the Apple. Normally, the card is placed in slot 4. Once the card has been installed, the Apple can function in either the 6502 or the Z-80 mode, and one can switch back and forth using software commands from the keyboard.

When in the Z-80 mode, the Apple II can process CP/M commands and execute BASIC programs utilizing the powerful MBASIC interpreter, Version 5.0 from Microsoft. The Z-80 Softcard is easy to install and takes only a few minutes of your time to begin experiencing the power of CP/M on the Apple. The installation instructions are easy to follow and are contained in Volume I, Chapter 1, of the Microsoft Softcard manual. If you have purchased some other brand of Z-80 card, refer to the documentation that came with the system.

Since the format of the booting diskette keys the Apple as to which

operating system to use, it is important to keep your Apple diskettes appropriately marked. One suggestion is to use a gummed label on the diskette marked with the information as to what is on it. If the diskette is CP/M formatted, place the letters "CP/M" on the label to identify it as such.

Recall from the latter part of Chapter 1 that you formatted blank diskettes using the CP/M FORMAT and COPY /S option. FORMAT and COPY/S, in effect, makes the diskette into an Apple CP/M "bootable" diskette. Thus, when the diskette is placed in drive A, the Apple computer will recognize the fact that the diskette is a CP/M system and activates the Z-80 card automatically. If the diskette is an Apple DOS 3.3 diskette, the Apple computer will automatically activate its own internal hardware system and boot into Applesoft BASIC.

At this point, it is useful to review how data is stored on the surface of diskettes and the organization of data storage on diskettes. This will help you to better visualize the interface between the diskette and the computer system and impress upon you the importance of keeping diskettes in a safe, secure environment.

## 2.3 A Look At Data Storage On Disks

Computer data starts with a character. A character is created by the computer using an 8-bit, 16-bit, or 32-bit binary code. A binary code is a series of zeros and ones. If a computer uses an 8-bit code, it is called an 8-bit machine. If it uses a 16-bit code it is called a 16-bit machine, etc. Your Apple II system is an 8-bit microcomputer.

A series of characters are stored in a field. A field may be used for the names of customers, for their addresses, and so forth. A series of fields designed to hold information of interest relating to a particular individual or company is called a record. A collection of records is called a file, and a collection of files is called a volume.

An example will help you to understand this concept. Suppose you are president of a club. You need a file containing name, address, phone number, zip code, amount of dues owed, and due date of dues payment for each member of your group. You have allocated a maximum of 24 characters to the field where a member's last name is written, ten characters to first name, 24 characters to address, fourteen characters to the city, two characters to the state, five characters to the zipcode, 12 characters for the phone number, 8 characters for the amount owed, and 6 characters to the due date. This total of 105 characters makes up the record of a single individual club member. Letters, numbers, spaces, and special symbols all are considered characters by the computer. You must be careful to include them in your count.

The collection of all names, addresses, etc., of the members of the club constitutes a data file. This data file is simply a collection of the individual records associated with each club member.

Now, suppose you wanted to store this data on a semi-permanent storage medium for future use. If you entered the data into the computer from the keyboard, the data would be stored in the computer's memory. But if you

turned off the power to the system, all data and programs stored in RAM would be destroyed. What you will want, is to transfer this data to a storage medium that will allow you to access it later. The most popular storage medium for microcomputers is the floppy diskette.

Floppy diskettes are made of thin circular pieces of plastic coated on both sides with magnetic film. To protect the diskette's surface, the plastic disk is enclosed in a stiff cardboard jacket (see Figure 2-1) to provide protection for most of the diskettes's surface. However, a portion of the surface is accessible via the read-write slot that is present on both sides of the diskette. Therefore, when handling diskettes, care must be taken not to touch the surface of the diskette, as fingerprints and scratches can damage the surface and partially destroy the coded information stored on the diskette.

In the center of the disk is a large circular hole called the spindle hole. This hole is used to center the diskette on the drive spindle inside the disk drive. When the disk drive light is on, the diskette is being rotated inside the jacket at speeds as high as 360 rpm. The exact rpm depends upon the design of the disk drive and must be within very precise tolerances. The drive motor is regulated by electronic elements to control this speed. When the disk drive door is closed, the drive spindle centers the diskette in preparation for activating the diskette's rotation.

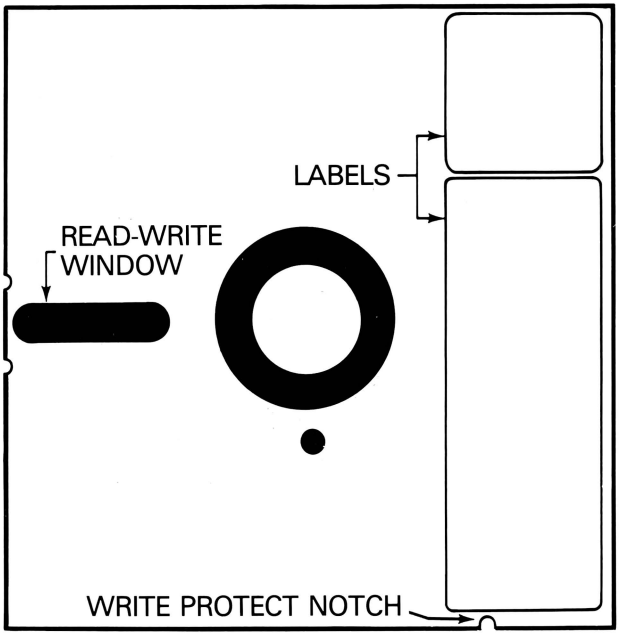



**Figure 2.1. Typical Floppy Diskette**

Further examination of Figure 2.1 reveals a small rectangular notch cut in the upper righthand side of the diskette. This notch is called the "write-protect notch" and protects the diskette from being written to by the system. If you want to protect a 5 1/4 inch diskette from being written to, simply place a piece of tape over the notch. When the notch is taped, the diskette is protected from disk-write operations. However, the system can still read the

diskette. When using 8-inch diskettes, the procedure is reversed. The notch must have a piece of tape over it in order to be written to. Once removed, the write- protect notch open, the 8-inch diskette is write-protected. It's a good idea to write-protect certain diskettes with tape. However, some of your diskettes will contain data files that must be read and then rewritten by the system. Be sure those diskettes are not write-protected or you will get a disk I/O error message on the screen.

The surface of the diskette is divided into "disk tracks" much like the grooves on the surface of a record. Unlike record grooves, which spiral to the inside of the record, the tracks on the diskette are concentric. Depending upon the type of system used, diskettes are formatted by the operating system to contain from 35 to 70 tracks per surface. Most CP/M systems format the diskette for 40 tracks. However, Apple CP/M formats the diskette to match the Apple DOS format of 35 tracks per surface.

Each track is divided into a "disk sector." Disk sectors are formatted to hold a predetermined number of bytes of data. The Apple DOS and Apple CP/M disk formatting procedure prescribes 256 bytes of data per sector. (Recall, one byte equals one character.) Further, the Apple formatting system provides for 16 sectors per track. Thus, an Apple formatted diskette can hold 35 tracks x 16 sectors/track x 256 bytes/sector or 143,360 bytes of data. Since 1Kb = 1024 bytes (by convention), then an Apple CP/M or Apple DOS formatted diskette can hold a maximum of 140Kb of information (143,360/1024). However, not all of the 35 tracks are available for disk storage. Certain tracks are reserved by the operating system to contain certain information needed by the operating system. As such, a typical Apple CP/M formatted diskette can hold approximately 126Kb of formatted information.

The most common CP/M disk format is 40 tracks per disk sector. The number of sectors per track vary from system to system along with the density of writing data to the diskette. Typical diskette capacities range from 96Kb to well over 360Kb per diskette surface. Figure 2.1 illustrates the track versus sector layout.

With the potential to store large amounts of information on the surface of the diskette, how does the system know where to find the particular data needed by the user? If you will refer to Figure 2.1 you will see a small circular hole just to the right of the larger spindle hole. This hole is called the "index hole." If you would gently grasp the inside edges of the spindle hole and slightly rotate the plastic diskette inside the cardboard jacket, you will notice a small circular hole (or holes) punched in the diskette. If the diskette contains only one hole, the disk is said to be "soft sectored." If the diskette contains many such holes, the diskette is termed "hard sectored." Some microcomputer systems (the Osborne for example), require the use of soft sectored diskettes only. Other systems require hard sectored diskettes. The Apple system can recognize either hard or soft sectored diskettes, thereby allowing the user to purchase and use either type. The purpose of the index hole for most systems is to allow a beam of light to pass through the hole and be sensed by the electronics built into the drive. This aids in locating the particular sector. This "pulse" of light can aid the system in determining "where" the read/write head is on the surface of the diskette at any time.

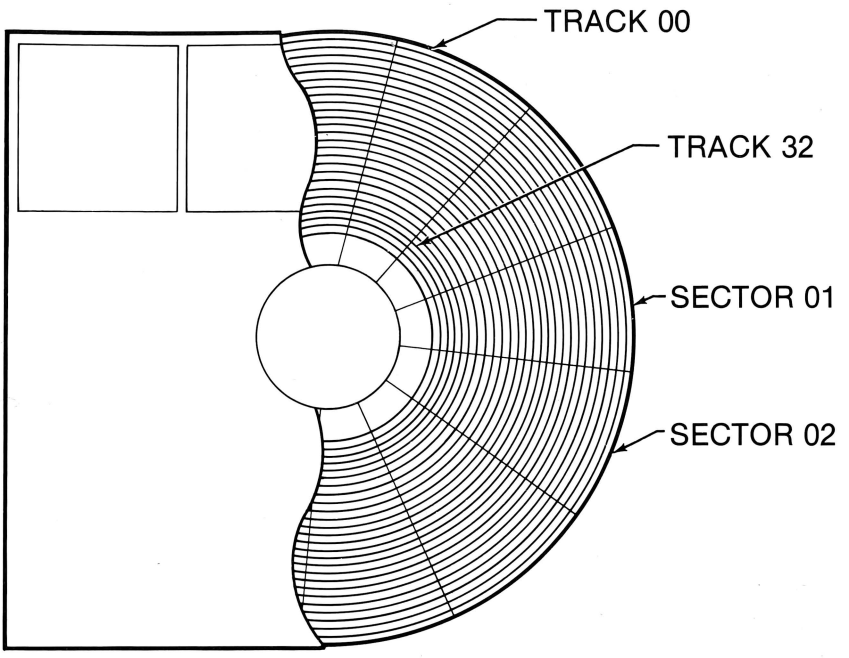



**Figure 2.2. Track and Sector Layout**

Apple CP/M formats its disks for 35 tracks instead of the more popular 40 track configurations used by other microcomputer systems (IBM, Osborne, Texas Instruments, etc.). The reason for this formatting difference is the fact that Apple DOS ( a non- CP/M system) formats diskettes in 35 tracks and Apple CP/M must be compatible. As a user of these diskettes, you do not have all 35 tracks available for data and/or program storage. The system must use certain specified tracks for storing the operating system and a very important entity called the "disk directory." The disk directory is sort of a cross reference file that aids the operating system in finding files that you specify and maintaining important information concerning those files. Each time a file is specified, the disk directory is accessed to help in determining where on the disk the file is located. If you add to or delete from a given file, the statistical data maintained for that file are updated accordingly. Therefore, out of approximately 143K bytes of total disk space, the user ends up with approximately 126K bytes of usable disk space per surface.

## 2.4 Starting Apple CP/M

The correct procedure for starting your Apple CP/M system is to place the diskette containing your operating system in drive A, turn on all peripherals, then turn on the computer. It is recommended that you install a strip power source with a switch and some electronic device for controlling power surges. The power strip will allow you to turn all parts of your computer system on at the same time and protect your system from potential damage.

During the process of starting the system, the control program is loaded

into the computer with the bootstrap program. The control program then takes over and performs a check of the unit. After this process is complete, the CP/M prompt character set "A>" will appear on the screen. This means that CP/M is ready for a CP/M command from the keyboard and the current disk drive is A.

## 2.5 Apple-CP/M Commands

Your Apple CP/M operating system consists of routines you do not see, and a selection of commands you can address to perform the functions required of the computer. These CP/M programs are stored on the Apple CP/M diskette and are present each time you boot with an Apple CP/M system master. The programs stored on the 16 sectored Microsoft version of Apple CP/M (CP/M 2.2) are:

```
APDOS        LOAD
ASM          MBASIC
CONFIGIO     PIP
COPY         STAT
DDT          SUBMIT
DOWNLOAD     XSUB
DUMP         CPM56
ED           GBASIC
FORMAT       RW13
```

It is important to point out that the Apple CP/M commands shown above are "custom" for the Apple II system and do not necessarily represent the set of CP/M commands found on other microcomputers. Digital Research and Microsoft developed a subset of CP/M to function on the Apple system consistent with Apple's 35-track disk formatting configuration. If you are accustomed to working on another CP/M system, you will find some different CP/M commands for the Apple. With a little reading in the Apple CP/M user's manual, you will be able to adapt to the slight changes with a minimum of difficulty.

Each of the above commands will be reviewed briefly to give you an overview of the command. Refer to the particular Apple CP/M manual for a detailed discussion of the commands.

1. APDOS
   APDOS is a utility program that will allow you to transfer program and/or data files from standard 35-track Apple II diskettes to CP/M diskettes. APDOS may be used to transfer text or binary files only, and requires at least two disk drives. For a detailed step-by-step discussion of how to use APDOS, refer to the Microsoft "Software Utilities Manual" that came with the softcard.

2. ASM
   ASM is the Apple CP/M 8080 assembler and can be used with the DDT utility to write and debug 8080 assembly language programs. Use of this utility requires that the user possess advanced knowledge outside the scope of this text. It is not required to run any of the programs contained herein.

3. CONFIGIO
The CONFIGIO program is a valuable utility used to configure the Apple CP/M operating system environment to your particular system. It has four major functions; to configure input and output for use of an external terminal, to aid in the redefinition of the keyboard characters, to load user input/output software, and to read and write the input/output configuration. The user of this utility must refer to the Apple CP/M manual for details concerning this utility.

4. COPY
The COPY utility is used to copy an entire CP/M diskette to a formatted CP/M diskette and to create blank CP/M system diskettes from a newly formatted diskette. Use of this utility was discussed in Chapter 1.

5. DDT
The DDT (Dynamic Debugging Tool) is a powerful CP/M utility used to perform interactive testing and debugging of 8080 assembly language programs. Only the experianced assembly language programmer will use this powerful utility. Details concerning DDT are contained in the Microsoft ''CP/M Reference Manual'', Chapter 5.

6. DOWNLOAD
DOWNLOAD is a CP/M utility that will enable the user to transfer files from one CP/M computer to the Apple II via a RS-232 serial data linkage. Use of DOWNLOAD requires a working knowledge of 8080 assembly language programming and is intended for experienced programmers.

7. DUMP
DUMP is used to display the contents of a disk file in hexadecimal form. Knowledge of how to intrepret hexadecimal listings is needed in order to use this utility.

8. ED
ED is the CP/M text editor and is used to create and edit CP/M files. ED allows the user to edit any CP/M generated text file but is extremely primitive. CP/M text file editing can better be accomplished by using a word processing language to create and edit text files.

9. FORMAT
The CP/M utility FORMAT is used to format a blank diskette so that it can accept data and program files. Since a freshly formatted diskette will not boot, (but will accept data files by using the PIP utility), use of the COPY/S option is required to place a copy of the current CP/M system on the diskette. FORMAT was discussed in Chapter 1.

10. LOAD
LOAD is used to convert a disk file with the extension .HEX (CP/M extensions will be discussed in the next section) into a machine-executable .COM file.

11. MBASIC
MBASIC is the Microsoft version of the BASIC language. MBASIC is required for the programs contained in this book and must

be resident on your program diskettes. MBASIC represents one of the most powerful versions of the BASIC language and supports low resolution graphics and sound and game controls. In addition, many BASIC commands not supported by Applesoft BASIC are contained in MBASIC. MBASIC may be PIPed onto your programs using the next CP/M utility.

12. PIP
PIP (Peripheral Interchange Program) is one of the most frequently used CP/M utilities. PIP is used to transfer files (one or more) from one diskette to another. It is also used to copy and append files and to transfer files from disk to terminal devices and printers. It is strongly suggested that the user become familiar with PIP, as it is needed to set up the files and programs described in this text.

13. STAT
STAT is a useful CP/M utility to provide you with general status information concerning diskette capacity, file sizes, file indicators, and device assignments. STAT may be used to alter file indicators and device assignments.

14. SUBMIT
The SUBMIT utility is used to allow CP/M commands and program input lines to be executed from a disk file as opposed to direct keyboard entry. This utility allows the user to use a preprogrammed disk file to perform ''automatic'' processing of CP/M files.

15. XSUB
The XSUB utility is used with SUBMIT to allow character input from a disk file at all times during execution of programs.

The remaining Apple CP/M files are contained on the 16-Sector Softcard diskette that came with the Softcard package. Other "softcard" systems may or may not contain these features.

16. CPM56
CPM56 is used to update a 44K CP/M system disk to a 56K system. This allows you to implement a full-sized 56K CP/M system on the Apple. It is essential that you design your system under the 56K format! Further, you must have a 16K language or RAM card in slot 1 of the Apple since CPM56 will not function on a 48K system.

17. GBASIC
GBASIC is the high resolution version of MBASIC and is contained on the Microsoft CP/M system master diskette. GBASIC is not required for the programs described in this text. GBASIC allows the user to create and run high resolution BASIC programs. Refer to the Microsoft ''BASIC Reference Manual'' for detailed information.

18. RW13
RW13 is a utility used to allow 16-sectored CP/M to access files on the older 13 sector diskette format. When used with PIP, RW13 is useful for transferring files from a 13-sectored to a 16-sectored diskette and requires two or more disk drives.

In addition to the CP/M utilities and language programs, there exist several CP/M "commands" that are useful. These commands are called "built-in" commands. They are:

`DIR, ERA, REN, and TYPE.`

DIR and TYPE have been covered. ERA and REN are summarized below.

ERA - The ERA command is used to erase files on the currently logged disk. You must be very careful in using the ERA command since it is possible to destroy valuable programs. In using the ERA command, DON'T DELETE ANY OF THE FILES ON THE CP/M MASTER DISKETTE. Examples of the ERA command are:
ERA B:TEMP.OLD - erases the file TEMP.OLD on drive B.
ERA *.* - erases all the files on the currently logged drive (A or B). If you attempt to erase all the files on a diskette, CP/M will ask the following:
ALL (Y/N)?
If you do not intend to erase all the files on a disk, respond with an "N".

REN - The REN command is used to rename CP/M files. The general form of REN is:
REN newname = oldname
where "newname" and "oldname" are file names. Examples are:
REN ACCOUNT.NEW = TEMP.OLD
The old file TEMP.OLD is renamed to ACCONT.NEW on the current drive.
REN B: TAX.DAT = TEMP.DAT

The old file TEMP.DAT on Drive B: is renamed to TAX.DAT on Drive B.

## 2.6 CP/M File Name Extents

CP/M provides a fairly restrictive syntax for naming files. For example, Applesoft allows the user to name files with a maximum of 30 characters in the file name. The first character of an Applesoft file name must be a letter, and any character that you can type from the keyboard except a comma can be part of a valid Applesoft file name. CP/M, on the other hand, has different rules regarding file names. The following rules apply to the naming of CP/M files:

1. Valid file names contain two elements, the file name and the file "extent." The file name and the extent are separated by a ".".

2. The file name portion cannot exceed 8 characters.

Examples:
TEST.BAS
MENU.BAS
TAX.DAT

3. The extension portion of the file name (which is optional in some cases) can consist of 3 characters after the ".". The three character extent aids the user in identifying the file type. One of the more common extents is the ".COM" extent. COM stands for COMMAND. Any file with an extension of .COM is a transient command file type and can be invoked by simply typing its file name without the extent. For example, to invoke MBASIC from the system master the following CP/M command can be used:
   A> MBASIC.COM <RETURN>
   or,
   A> MBASIC <RETURN>
   Either method will work since MBASIC.COM is a transient command file.

The following extents are suggested when working with CP/M files:
.BAS For BASIC programs
.BAK For backup copies of a file
.BAT For batch mode operation (reviewed as part of AUTOEXEC.BAT)
.COM Command file
.DAT For Data files
.EXE External file
.HEX An ASCII representation of a command file
.MAP For linking files
.OBJ Object Modules -- for linking files
.$$$ Working files

## 2.7 Summary

This chapter has attempted to highlight some of the basics of Apple CP/M. However, the serious user will find it necessary to refer to one of the many fine softcover textbooks covering CP/M. The user must remember that Apple CP/M has some features not found in other 8-bit CP/M versions. Therefore, reading of the CP/M manual or reference that came with the particular Z-80 card is essential. It is suggested that the first-time user of Apple CP/M spend several hours becoming familiar with the Apple CP/M version before running the software described in this book. The time investment will be well spent.

In summary, Apple CP/M is a very powerful operating system with many capabilities, some of which are needed only for specialized functions. As a general user, it will take many months before you use all the utilities, but you should be aware of what is available so you can use the operating system when needed.

The next chapter will begin the introduction to the Apple II CP/M accounting package which will allow you to set up a workable, inexpensive accounting system to aid in the management and control of a small- to medium-sized business. In the process of learning about the accounting system, you will also discover that you will be increasing your computer knowledge and having fun at the same time.

# References

CP/M REFERENCES:

*OSBORNE CP/M USER GUIDE* by Thom Hogan, Osborne/McGraw-Hill Publishing Co., Inc., 630 Bancroft Way, Berkeley, CA. 95710, 1981.

*BASIC-80 and CP/M* by Jack Jay Purdum, Macmillan Publishing Co., Inc., New York, N.Y., 1983.

*MICROSOFT SOFTCARD USER GUIDES (Vols I and II)*, Microsoft Consumer Products, 400 108th Ave. NE, Suite 200, Bellevue, WA 98004, 1980.

*COMPUTER STATION'S PROGRAMMER'S HANDBOOK to the Apple II*, Computer Station, 11610 Page Service Drive, St. Louis, MO.,63141 (314) 432-7019, Edited and compiled by Conway B. Christensen, 1981.

*INSIDE CP/M—A GUIDE FOR USERS AND PROGRAMMERS* by David E. Cortesi, Holt, Rinehart and Winston, New York, N.Y., 1982.

*USING CP/M ; A SELF TEACHING GUIDE* by Judi N. Fernandez and Ruth Ashley, John Wiley & Sons, New York, N.Y., 1980.

*MASTERING CP/M* by Alan R. Miller, SYBEX , Inc., Berkeley, CA., 1983.

*THE CP/M HANDBOOK (with MP/M)* by Rodnay Zaks, SYBEX, Inc., Berkekey, CA., 1982.

# II: ACCOUNTING

# 3. A Collection Of Business Programs

This section presents a collection of seven business programs that may be used to maintain the accounting system for small- to medium-sized businesses. Included in the collection are programs that will enter a periods transaction, help balance a checkbook and produce a series of mailing envelopes or labels, a business invoice and check, an income and expense statement, and a balance sheet.

The series operates from a menu program that controls the flow of the activities. Programs included in this series and their functions are:

| Program | Function |
|---|---|
| RUN"x:MENU"<br>(Where "x" denotes drive A or B) | The menu program. |
| (C)heckwriting program | Checks are produced from both the keyboard and a transaction file. A transaction file also may be produced if input is from the keyboard. |
| (I)nvoice program | Produces an invoice from the keyboard and a transaction file if desired. |
| (R)eport:Income/expense | Produces an income and expense statement, balance sheet statement, and a balance sheet. |
| (S)etup chart of accounts | Used to set up the chart of accounts and other initialization information needed to operate the business system. |
| (T)ransaction program | Inputs the periodic business transactions, aids in reconciliation of the business cash account, and produces hard copy as desired. |

Programs are listed in order as they appear on the menu program, with the exception of the menu program. It is accessed as follows: RUN"x:MENU"<CR> is the method used to call the program the first time. The x: represents the disk drive where the program is. If MENU is in the logged drive (normally A), the "x:" may be omitted. If the diskette with MENU on it is in B drive, call it as RUN"B:MENU"<CR>.

To initiate the accounting package, place the program master diskette that contains a copy of the Apple CP/M system in drive A, close the drive latch, and turn the Apple on. When the CP/M prompt appears, type MBASIC <CR>.

In a few seconds, the Microsoft MBASIC logo will appear in the upper lefthand corner of the screen. Just below this logo will appear the flashing rectangular cursor that indicates the MBASIC system is ready to receive MBASIC commands. It is important to point out that when you are in the MBASIC mode the CP/M commands do not work! Figure 3.1 illustrates the Microsoft version of MBASIC initial screen.

```
BASIC-80 Rev. 5.2
[Apple CP/M Version]
Copyright (C) 1980 by Microsoft
Created: 12-Nov-80
26483 Bytes Free
Ok
```

**Figure 3.1: Initial MBASIC Screen from CP/M**

At this point, we will introduce one of the differences between CP/M commands and MBASIC commands. If at any time you want to display your diskette files from MBASIC, you type:

```
FILES <CR>
```

As you recall, CP/M required the command "DIR <CR>" to display files on the current drive. The DIR command will not work in MBASIC and is replaced by the command FILES <CR>. Figure 3.2 illustrates what you should see on the screen when you have the accounting diskette in drive A.

```
BASIC-80 Rev. 5.2
[Apple CP/M Version]
Copyright (C) 1980 by Microsoft
Created: 12-Nov-80
26483 Bytes free

Ok
FILES
MBASIC.COM      CHART.BAS      TRANS.BAS      CHECK.BAS      MAIL.BAS
CPA.BAS         MENU.BAS
OK
```

**Figure 3.2: Apple CP/M Screen Results For FILES Command**

Therefore, use of the FILES command in MBASIC serves the same function as DIR does in CP/M.

To initiate the accounting package, perform the following steps:

1. Place an Apple CP/M version of the accounting master in drive A and turn on the power to the Apple. (REMEMBER TO USE A BACKUP COPY AND STORE THE MASTER COPY.)

2. Type from the keyboard the following:
   MBASIC <CR>
3. When the MBASIC flashing prompt appears, type:
   RUN "MENU <CR>

MBASIC will obtain the MENU.BAS file from the diskette and load it into RAM. As soon as the program is loaded, the system will RUN (execute) the MENU program. Section 3.3 of this chapter will cover this program.

The next section will review aspects of basic accounting but is not intended to be a detailed coverage. For those readers who know very little about the mechanics of accounting, several texts and manuals exist to assist you. Your local bookstore and community education programs may be consulted for help in this area. When presenting the accounting programs, we assume that the user has a fundamental working knowledge of accounting.

## 3.1 Accounting Knowledge

Our accounting system program is based on what is known in the business world as the double entry system. It is the system used throughout the United States. In this system two columns of numbers are maintained. The totals of the two columns are checked against each other and must, at all times, be in balance.

When a number is placed in the lefthand column, it is called a "debit." When a number is placed in the right column, it is called a "credit." The sum of the debit entries and the credit entries must be exactly the same. In most simple transactions the debit and credit entries are exactly the same. When you are making an entry that involves several different accounts, such as the payment of a mortgage, then the sum of the debit entries must be equal to the sum of the credits. A typical mortgage payment might involve the following transactions:

| Post to account of | Debit | Credit |
|---|---|---|
| Cash | | Total amount of payment |
| Interest expense | | Amount of interest |
| Reduction of loan | | Amount loan reduced |
| Escrow | | Amount of escrow payment |

The word "post" means to record the transaction (the event) in the accounting books. In the computer system, the act of posting is the task of typing the information into the computer. It is possible to ask the computer to check the sum of the debits and credits after each transaction. When first starting it may be a good idea to use the sum checking ability often to make sure you are not making errors.

There are 2 accounting reports produced by the general ledger part of the program. These reports are the income and expense statement and the

balance sheet. The first report is often called a "Profit and Loss statement" (P&L Statement).

The "Income and Expense statement" tells you how much you earned or lost for a specific period. This file will always be different by the amount of the period's earnings. When period's earnings are taken into consideration, then this report must also be balanced. Our computer files may look out of balance at times but are actually in balance when the profit or loss is considered.

The balance sheet is designed to tell you where your organization is financially at any given time. It is like a snapshot of the organization's financial condition. As its title indicates, the balance sheet must always balance.

We recommend that you work closely with an accountant to make sure your system is consistent with his or her way of doing business so your accounting costs may be kept to a minimum. Coordination with your accountant is most important during the initial system organization and setup.

## 3.2 Organization Of This Section

Programs have been listed in alphabetic order in the MENU to aid in selection during normal use. The order in which you use the programs to set up and operate a business is the order of the chapters in this section. This order is:

| Chapter | Program |
|---|---|
| 4 | (S)etup chart of accounts |
| 5 | (T)ransaction program |
| 6 | (R)eport:income/expense and balance sheet |
| 7 | (I)nvoice program |
| 8 | (C)heckwriting program |
| 9 | (T)ransaction—Using the transaction program to balance your cash account. |

The operation and design of the MENU program will be examined in this chapter. In each additional chapter, the program under consideration will be examined from both user and programmer points of view. The objective is to provide a program you can use as is and at the same time provide a detailed look at a program that may be customized for your specific needs. The exception to this procedure will be Chapter 9 when we will examine how to use the transaction program for check balancing.

One program (Reports: Income/Expense and Balance Sheet) requires that up to 4 files to be open at the same time. In Chapter 6, we will instruct you

on how to load and run this program since it will require that 6 data files be active at one time. Normally, MBASIC permits 3 files to be active. Therefore, we will have to specify, in advance, that we will require 6 files. This is the only program in the set that requires some "special attention" when loading. File input/output and internal storage limit the size of the business that may be handled by our programs. The exact capacity limitations depend on the size of the chart of accounts, or the size of the transaction file, and how they interact with each other. In terms of transactions, as many as 300 have been input using a 64 KB memory system. Some companies with a 100-unit chart of accounts and 64 KB have not approached the limit of the system.

## 3.3 SYSTEM OVERVIEW

All programs in this collection are related in that they use or produce a set of files used and updated by the other programs in the system. The transaction program produces a transaction file that can be used for input to the checkwriting routine to produce checks or as part of the input to the report writing routine to produce the income/expense statement and the balance sheet.

A transaction file following the same format may be produced by both the invoice and checkwriting programs. These transaction files, once produced, may be brought into the transaction program for update and correction and then used as input for other routines.

Care should be taken in the use and naming of the transaction files. It is recommended that you use APRIL, MAY, ... (the name of the month) if you are using monthly files or WEEK1, WEEK2, ... if you are using weekly files. Take care not to use more than 7 characters in a name. Months with long names should be abbreviated.

In the startup routines, two files are created: a current balance sheet file and an income/expense file. These files are used as input to the report generation program and, with additional files plus the transaction files, produce a current income/expense and balance sheet report and new income/expense and balance sheet file for the following month. Name these files with the month or other time period to which they refer. If you name one JUNE, the income and expense file will be called JUNE.INC and the balance sheet file will be called JUNE.BAL.

Table 3-1 details the relationship between the various parts of the accounting system. If you are interested in redesigning the system for a custom application, you should examine carefully these relationships. If you are planning to use the program without changes, you need only a rough understanding of the files and their relationships.

Table 3-1 lists the files used by our system and the programs in which the files are used. Table 3-2 gives a brief description of each of the files and what they are used for.

Table 3-1. Files and Programs

| Files | Menu | Break | Setup | Trans | Report | Invoice | Check | Mail |
|---|---|---|---|---|---|---|---|---|
| Trans-action filespec .BAL | no | no | no | yes | yes | yes | yes | no |
| filespec .INC | no | no | yes | no | yes | no | no | no |
| NAMEX. DAT | no | no | yes | no | yes | no | no | no |
| ACCT. NUM | no | no | yes | no | yes | yes | yes | no |
| DATAF. INF | no | no | yes | no | yes | no | no | no |
| INVNUM. DAT | no | no | yes | no | yes | no | no | no |
| COMPANY. DAT | no | no | no | no | no | yes | no | no |
| NOCHECK. DAT | no | no | no | no | no | yes | no | no |
| SUPPLY. DAT | no | no | no | no | no | no | yes | no |
| For Printer | no | no | no | no | no | no | yes | no |
| | no | no | yes | yes | yes | yes | yes | no |

Table 3-2 Files and Their Uses

| File | Use |
|---|---|
| 1. Transaction | These files store and transfer the transaction information between runs and programs. |
| 2. filespec.BAL | These files store the last period's balance sheet information for next period's use. |
| 3. filespec.INC | These files store the last period's income/expense information. |
| 4. NAMEX.DAT | These files contain your company's name and address. |
| 5. ACCT.NU | These files contain the number of the last asset account, last liability account, last stockholder's equity account, last income account, and last expense account. |
| 6. DATAF.INF | These files contain the number of title accounts for report printing. |
| 7. INVNUM.DAT | These files contain invoice number, the accounts receivable number, and sales tax. |
| 8. COMPANY.DAT | These files contain customer lists. |
| 9. NOCHECK.DAT | These files contain check numbers. |
| 10. SUPPLY.DAT | These files contain suppliers lists. |
| 11. for Printer | These files aid in changing print size on the MX-80 printers. |

## 3.4 RUN "x:MENU"

To run this program you must know which disk drive the program is in. Use a DOS diskette which contains the system programs in drive A. All extra BASIC programs should be removed. Place an Apple CP/M formatted diskette in drive B to store the data for your company. By following this practice, you will be able to use the program set for several different businesses. Each business will have its own data diskette. You should be able to store close to a year's data on one of your 127 KB diskettes for most small- to medium-sized organizations.

The MENU program starts with the following credits and question:

```
APPLE CP/M ACCOUNTING SYSTEM
Version 83.1
Steven Zimmerman, Ph.D.
Leo M. Conrad
Donald R. Smith, Ph.D., P.E.
(Press <CR> to display Main Menu)
```

Accounting System Menu:

```
APPLE CP/M ACCOUNTING SYSTEM
*** MAIN MENU ***
Select one of the following:
(C)heckwriting program
(I)nvoice program
(M)ailing label program
(R)eport: income/expense & balance sheet
(S)et up chart of accounts program
(T)ransaction program
(Enter C/I/M/R/S/T <CR>)
```

**Figure 3.3: Screen Display for MENU**

The use of drives A and B are recommended for systems with two drives. For those of you with one, three, or four drives, the package may still be used with minimum effort by specifying where the programs are located in the above question. You must have all programs in the same disk drive and all the data information in a second disk drive. In a very small system, it is possible to store both programs and data on the same diskette. If you have double-sided drives, it is possible to use a single diskette for both data and programs. However, it is not recommended.

The program is designed to accept either upper- or lowercase answers to most questions. You may answer some questions with only the first letter of the word requested. The exceptions to this rule are made to protect you from making costly errors. For example, when asked if you wish to initialize your file, you must answer with the word YES in uppercase letters before the system will do as instructed. If you initialize your file in error, it will be very

costly in terms of information and time because all data stored in that file will be lost.

Assume that you have the MENU screen up as shown in Figure 3.3. The first program of interest is the Setup program (see Chapter 4 for details concerning (S)etup). To execute this program from the MENU program, enter the letter "S" from the Apple keyboard followed by <CR>. The MENU program will immediately access drive A and run the Setup program for you. All of the other programs are accessed in the same manner except the (R)eport program. Special instructions for running this program will be given in Chapter 6. *Do not attempt to run the (R)eport program from the MENU until you have read Chapter 6!*

*WARNING:*

For those of you who are copying the programs out of the book, **BE SURE TO SAVE THE MENU PROGRAM BEFORE YOU EXECUTE IT!** If you do not do so, the program code will be lost each time you execute the program.

## 3.5 Examining The Program

In line 10 of each program, you will find the program name. In the menu program this line reads:

```
10 REM SAVE ''MENU''
```

This may seem a strange way to name a program, but in Microsoft BASIC it is possible to record over an old program and lose valuable information. To minimize errors, list line 10 using the LIST 10<CR> and then use the screen editing capability to delete the line number and the word REM. Pressing <CR> results in the program being saved.

REMarks, like all other code, use room inside your computer. If you have capacity problems, one thing you can do to increase the capacity of your system is to remove all REMark statements. They are put into programs to help you understand what is going on. They do not change the operation of the program in any way.

In line 20, "HOME" clears the screen and the first screen display is printed. Lines 40-70 print the menu options on the second screen. The INPUT statement in line 70 requests the user to input one of the following letters:

```
C - to run the checkwriting program
I - to run the invoice program
M - to run the maillist program
R - to run the report generation program
S - to run the setup chart of accounts program
T - to run the transaction program
```

Lines 90-140 check on the value of the user input. If the user does not enter a C,I,M,R,S, or T, the program rings the "bell" (to call your attention to the screen) and makes you repeat the sequence again. In line 120, if you request the REPORT program (by entering an "R" or "r" from the keyboard), the program will branch to line 160. In lines 160-200, special instructions are given regarding loading and running this program.

In all the IF statements, (lines 90-140) the leftmost character alone is looked at to determine the action to be taken by the program. Note also that if your Apple has lowercase capability, you can enter either the upper- or lowercase version of the option.

## 3.6 Summary

The program series presented is part of an integrated accounting system for a small- to medium-sized company. The system assumes that the user has a foundation in accounting but little or no computer knowledge. Input of data is efficient as long as the user is knowledgeable in the task of posting to accounts.

For individuals interested in using the system as designed, no knowledge of the programs is needed except how to call the program desired. All individuals should study the files used by the program just to have an idea of how things work.

For those who wish to customize the programs, we recommend careful study of the filing system used and the relationships of the various programs in our program set. The task of redefining files and their relationships is very time consuming and complex because of the many different programs interfacing with the files. Some experience in BASIC programming is recommended before this task is started.

The organization of the chapters in this section follows the selections found in the menu program. In each chapter the user is walked through each program and an explanation is given of how it is supposed to operate . Then we review the manner in which the program is written. This procedure is applied to each successive program.

The next chapter presents the SETUP program and is extremely important to the proper functioning of the entire system. We suggest that the user follow carefully the example data presented in this chapter and go through the step-by-step instructions relating to the example before setting up the accounts for your own particular needs.

## 3.7 PROGRAM LISTING FOR MENU

```
10 REM ''MENU''
20 HOME:VTAB 2:HTAB 26:PRINT''APPLE CP/M ACCOUNTING SYSTEM'':
   PRINT:PRINT:HTAB 34:PRINT''Version 83.1''
30 PRINT:PRINT:HTAB 29:PRINT''Steven Zimmerman, Ph.D.'':
   PRINT:HTAB 34:PRINT''Leo M. Conrad'':PRINT:HTAB 29:
   PRINT''Donald R. Smith, Ph. D., P.E.'':VTAB 21:HTAB 23: IN-
   PUT ''(Press < CR > to display Main Menu)'';DM$
40 HOME:VTAB 2:HTAB 26:PRINT''APPLE CP/M ACCOUNTING SYSTEM'':
   PRINT:HTAB 32:PRINT''*** MAIN MENU ***'':PRINT:PRINT:
50 HTAB 8:PRINT''Select one of the following:'':PRINT:HTAB 21:
   PRINT''(C)heck writing Program'':PRINT: HTAB 21:
   PRINT''(I)nvoice Program'':PRINT:HTAB 21: PRINT''(M)ailing
   Label Program'':PRINT:
```

```
 60 HTAB 21:PRINT''(R)eports Income/Expense & Balance Sheet'':
    PRINT:HTAB 21:PRINT''(S)et Up Chart of Accounts Program'':
    PRINT:HTAB 21:PRINT''(T)ransaction Program'':PRINT:PRINT:
 70 HTAB 30:INPUT ''(Enter C/I/M/R/S/T ) ''; SL$
 80 DD$=''A''
 90 IF LEFT$(SL$,1)=''C'' OR LEFT$(SL$,1)=''c'' THEN GOTO 210
100 IF LEFT$(SL$,1)=''I'' OR LEFT$(SL$,1)=''i'' THEN GOTO 160
110 IF LEFT$(SL$,1)=''M'' OR LEFT$(SL$,1)=''m'' THEN XX$=DD$+'':
    MAIL.BAS'':RUN XX$
120 IF LEFT$(SL$,1)=''R'' OR LEFT$(SL$,1)=''r'' THEN GOTO 160
130 IF LEFT$(SL$,1)=''S'' OR LEFT$(SL$,1)=''s'' THEN XX$=DD$+'':
    SETUP.BAS'':RUN XX$
140 IF LEFT$(SL$,1)=''T'' OR LEFT$(SL$,1)=''t'' THEN XX$=DD$+'':
    TRANS.BAS'':RUN XX$
150 PRINT CHR$(7):HTAB 10:PRINT''INVALID ENTRY...RE-ENTER'':GOTO
    70
160 HOME:PRINT CHR$(7):VTAB 5: PRINT''WARNING!...THIS PROGRAM RE-
    QUIRES 4 FILES TO BE OPEN''
170 PRINT:PRINT''YOU MUST RELOAD MBASIC WITH THE FOLLOWING COM-
    MAND FROM CP/M'':PRINT
180 PRINT'' FROM CP/M, TYPE THE FOLLOWING:''
185 IF LEFT$(SL$,1) = ''I'' OR LEFT$(SL$,1)=''i'' THEN PRINT:
    PRINT:PRINT'' MBASIC INVOICE/F:4'':GOTO 200
190 PRINT:PRINT:PRINT'' MBASIC REPORT/F:4''
200 PRINT:PRINT:PRINT''To return to CP/M type SYSTEM <CR>'':GOTO
    300
210 HOME:PRINT CHR$(7):VTAB 5: PRINT''WARNING!...THIS PROGRAM RE-
    QUIRES 4 FILES TO BE OPEN''
220 PRINT:PRINT:PRINT''YOU MUST RELOAD MBASIC WITH THE FOLLOWING
    COMMAND FROM CP/M'':PRINT
230 PRINT:PRINT:HTAB        10:PRINT''MBASIC        CHECK/F:4'':
    PRINT:PRINT:PRINT:PRINT''To return to CP/M type SYSTEM <CR>''
300 END
```

## References

*FUNDAMENTAL ACCOUNTING PRINCIPLES* by William W. Pyle and John Arch White, R. D. Irwin, Inc., Homewood, ILL., 1963.

*ACCOUNTING FOR MANAGEMENT CONTROL - AN INTRODUCTION* by Charles T. Horngren, Prentice-Hall, Inc., Englewood Cliffs, NJ.,1974.

*FINANCIAL STATEMENT ANALYSIS* by George Foster, Prentice-Hall, Inc., Englewood Cliffs, NJ., 1978.

*BASIC ACCONTING AND COST ACCOUNTING* by Eugene L. Grant and Lawrence F. Bell, McGraw-Hill Book Company, New York, N.Y., second edition, 1964.

# 4. (S)etup Chart Of Accounts

Accounting systems consist of data files that are updated periodically and then used to produce business reports. One necessary task is the job of originating the form and the values in these data files. This origination defines the structure of your accounting system, and the decisions you make at setup affect all of your future work. The task of setup requires you to think ahead to the future tasks of your system.

In the introductory remarks to this section, the 10 files needed to run the business system were listed. In some cases, it is possible to initialize the files as you use them. The following accounts must be initialized and created before the system may be used for the first time:

```
filespec.BAL
filespec.INC
NAMEX.DAT
ACCT.NUM
DATAF.INF
```

The program CHART.BAS is designed to do this setup. The program is called with the name (S)ETUP chart of accounts. All the files listed may be started with this program, and updating of NAMEX.DAT, ACCT.NUM, and DATAF.INF is performed with this program. This program is the "hub" of the accounting system. The firsttime user should study this chapter carefully and experiment with the example problem that is provided. It is suggested that you go through the example problem several times, setting up the required files more than one time. Before you begin setting up the system for your business, create some example files and run through the setup procedure with "dummy" data in order to gain confidence in the (S)ETUP operation. For those users with some familiarity with BASIC, the last section of this chapter describes some details of the program.

In filespec.INC, the yearend task of setting all the year-to-date information equal to zero is performed by CHART.BAS. The word "filespec" refers to the name you give to a particular file. This name should be seven characters or less. The words to the right of the decimal point are referred to as extenders, The extender "INC" refers to the fact that the file contains a year-to-date listing of the income/expense chart of accounts. The only time you must add the extender yourself is during the initial set up.

This chapter is organized into two principal parts. Part I is a review in detail of how the program works. Part II is a discussion of the program itself and what the statements do. If you are interested only in operating the program, study Part I in depth and spend a minimum of time reading about the details of the program. If you are planning to rewrite the program, then Part II must be studied.

# PART I

## 4.1 Some Accounting Background

Positive assets have debit balances and positive liabilities are credit balances. Assets may be divided between current assets and fixed assets. Current assets are cash, accounts receivable, inventory, and short term notes. Fixed assets are things that cannot readily be turned into cash, such as automobiles, buildings, and machinery.

Liabilities may be divided between short and long term. Short term liabilities are those which must be paid off within a year, while long term are those that will go over a year. Examples of short term liabilities are accounts payable, payroll payments, and taxes. Long term liabilities include mortgages.

In addition to assets and liabilities there is one more main classification of accounts in your chart of accounts. This classification is called stockholders' equity. The accounts in this classification include capital stock, the money originally invested in the company, retained earnings or surplus and all of the money ever earned by the company and reinvested into its operation.

## 4.2 Overview Of Numbering System For Chart Of Accounts

Account numbers may start at 0 and go as high as 32,000. All assets must be placed in the lowest number series, liabilities in the next higher number group, owners' equity in the next, income in the next, and expenses in the highest group. Selective accounts may be used as title accounts on the income/expense and balance sheet report. All account numbers must be integers, i.e., they cannot have special characters and may not contain decimal points.

One important task in designing your system is the defining of account numbers. Listed below are those used in our sample run:

| Type of Account | From | To | Highest No. Used |
|---|---|---|---|
| Current assets | 1 | 999 | ... |
| Fixed assets | 1000 | 1999 | 1901 |
| Short term liabilities | 2000 | 4999 | ... |
| Long term liabilities | 5000 | 5899 | 5000 |
| Stockholders' equity | 5900 | 5999 | 5999 |
| Income | 6000 | 8900 | 6900 |
| Expenses | 9000 | 9999 | 9650 |

**TABLE 4.1: Account Number Assignments for Example Run**

The highest number used is listed because the computer needs to separate accounts into their classifications, and this is the manner in which it is done. You may use any numbering system you wish, as long as the accounts are organized in the following ascending order:

1. **ASSETS**
2. **LIABILITIES**
3. **STOCKHOLDERS' EQUITY**
4. **INCOME**
5. **EXPENSES**

You also may have any number of title accounts to aid in the presentation of the reports. The title accounts we will use for our example run are:

| Number | Account |
|---|---|
| 2000 | Liabilities |
| 5900 | Stockholders' equity |
| 6000 | Revenue |
| 9000 | Expenses |

Most accountants have numbering systems they prefer to use. If you have been working with an accountant for a period of time he or she will already have a system worked out for you. To minimize costs, the use of the manual numbering system is recommended, if possible, within the guidelines we have listed.

Accounts should be input in numerical order, as this is good accounting practice and the program may have problems operating if such a practice is not followed.

## 4.3 Running The Program

To run the program, place a copy of the Apple CP/M program diskette in drive A and an Apple CP/M formatted diskette in drive B. The diskette in drive B will be used for storing your data files. Be sure that the drive door latches on both drives are closed. Turn on the power to the Apple. After the Apple CP/M prompt appears, type the following:

```
MBASIC <CR>
```

In a few seconds, the MBASIC prompt will appear. Type the following:

```
RUN ''MENU <CR>
```

The MENU options will appear on the screen as described in Chapter 3. To invoke the (S)ETUP program, type the following:

```
S <CR>
```

When calling the (S)ETUP option in the menu program, you will see on the screen:

```
CHART OF ACCOUNTS MAINTENANCE PROGRAM
by
Steven M. Zimmerman, Ph.D
Leo M. Conrad
Donald R. Smith, Ph.D., P.E.

*** (PRESS <CR> TO BEGIN) ***
```

**Figure 4.1: Screen One of the (S)ETUP Program**

After entering <CR>, the second screen of (S)ETUP appears as:

```
USER DIRECTIONS
1. Naming files:
Valid file names: max. of 8 characters followed by a period and 3
alpha characters
EXAMPLE: FIRST.BAL
Valid extent codes are:
.BAL .INC .DAT .NUM .INF
Bal. sheet or income/exp file name= ?
```

**Figure 4.2: Screen 2 of (S)ETUP Program**

The first question asks for the name of the file, filespec.BAL, or filespec.INC being updated or created. We will use the name FIRST.BAL for this sample run. It is best for the first part of the name to refer to some time period for which the balance sheet is being developed, in this case the first period. The second part of the name must have the letters .BAL or .INC. .BAL refers to balance sheet and .INC refers to an income/expense statement. For our example run, respond to the first screen prompt as follows:

```
Bal. sheet or income/exp file name= FIRST.BAL <CR>
```

After you answer the above question, the program continues:

```
Disk (A,B,C,D)? B
```

Place all programs on a system diskette in drive A and all data files on a formatted diskette in drive B. All examples are built upon the assumption that this policy is being followed.

The next question is:

```
Maximum number of items in the Chart of Accounts? 200 <CR>
```

This question is used to set the dimensions used for the variables internal to the program. It refers to the maximum number of items being placed in

memory at this time. If you have additional items in the income/expense statement, don't panic. It is not a problem at this time. Because of the varying amounts of memory in different users' machines, we made this value a variable under your control. Using this approach, you are able to adjust the system to fit your needs without making any program changes. Two hundred was chosen, because it was within the capacity of standard systems and has more than enough capacity to handle the sample application.

The next question relates to the inputting of the file:

```
Select input from (D)isk or (K)eyboard ? K <CR>
```

Since no file exists, the only method available to get the information into the computer is from the keyboard. The letter "K" or "k" is used to denote this choice. The computer is programmed to select the first letter from the material input and direct the flow of the program on the basis of this information.

The next activity is to start entering the information for the chart of accounts. Shown below is the series of answers given to this question group for our example run:

| Number | Acct. | Account | Debit | Credit |
|---|---|---|---|---|
| 1 | ?1 | ?Checking account | ?1000.00 | ?0.00 |
| 2 | ?2 | ?Petty cash | ?120.00 | ?0.00 |
| 3 | ?100 | ?Escrow | ?2209.50 | ?0.00 |
| 4 | ?500 | ?Accounts receivable | ?0.00 | 0.00 |
| 5 | ?1000 | ?Fixed assets | ?0.00 | ?0.00 |
| 6 | ?1001 | ?Computer equipment | ?27928.62 | ?0.00 |
| 7 | ?1002 | ?Res for depreciation | ?0.00 | ?12725.66 |
| 8 | ?-1 | | | |

There is a question mark by each number or account name you are expected to input. This means you must type the information and press <CR> after each item. The series can be terminated only at the question seeking another account number. Answer this question with a negative number to perform this job. It is assumed you know the value of the different items in your chart of accounts from your manual system.

After the input activity has been terminated, you will see the main action menu of this program on your screen:

```
Select (A)dd,(C)hange,(D)elete,(E)nd,(I)nsert,(L)ist,(M)erge (N)o
       Assets/Liab,(P)rint,(R)ecord,(S)um debits/credits, (T)itle
       Accounts,(V)alues change,(X)company's name,
                              (Z)ero all values
                              ENTER your choice?
```

During the use of this program, you will return to this menu again and again. Discussion will consist of a review of the alternate actions that various selections from this menu result in. As an example, enter "R" (record) to answer the above question followed by a <CR>.If you are working along with this

text, notice that the light for disk drive B turns on after you enter a <CR>. You will hear the disk drives operate as the information is being saved. After the disk save action is complete, you will return automatically to the program menu:

```
Select (A)dd,(C)hange,(D)elete,(E)nd,(I)nsert,(L)ist,(M)erge (N)o
       Assets/Liab,(P)rint,(R)ecord,(S)um debits/credits, (T)itle
       Accounts,(V)alues change,(X)company's name,
                              (Z)ero all values
                              ENTER your choice?
```

This time, answer the menu question with the letter "L" to obtain a listing of the information in the file. After pressing <CR>, the next statement you see and the sample answers are:

```
INPUT THE LINE NUMBERS TO BE LISTED
STARTING LINE NO.        ENDING LINE NO.
    ?1   <CR>            ?10 <CR>
```

For the data just entered, we wish to view the first ten entries starting with entry number 1 and terminating with entry no. 10. However, we made only seven data entries and the last entry (no.8) was a "−1" to indicate the end of entry. Entries 9 and 10 will appear as zero values. The following results will be displayed on the screen:

| Number | Acct. | Account | Debit | Credit |
|---|---|---|---|---|
| 1 | 1 | CHECKING ACCT | 1,000.00 | 0.00 |
| 2 | 2 | PETTY CASH | 120.00 | 0.00 |
| 3 | 100 | ESCROW | 2,209.50 | 0.00 |
| 4 | 500 | ACCTS REC. | 0.00 | 0.00 |
| 5 | 1000 | FIXED ASSETS | 0.00 | 0.00 |
| 6 | 1001 | COMPUTER EQUI | 27,928.62 | 0.00 |
| 7 | 1002 | RESV FOR DEPR | 0.00 | 12,725.66 |
| 8 | −1 | | 0.00 | 0.00 |
| 9 | 0 | | 0.00 | 0.00 |
| 10 | 0 | | 0.00 | 0.00 |

```
Select (A)dd,(C)hange,(D)elete,(E)nd,(I)nsert,(L)ist,(M)erge (N)o
       Assets/Liab,(P)rint,(R)ecord,(S)um debits/credits, (T)itle
       Accounts,(V)alues change,(X)company's name,
                              (Z)ero all values
                              ENTER your choice?
```

A debit value of 0.00 for account number 500, "Accounts receivable", was input. The value should have been $15,000. To make this correction, type V as the answer to the menu question (Values change) and the following questions and answers result:

```
INPUT line number to be changed? 4 <CR>
Number    Acct No.    Account                Debit          Credit
 4          500     Accounts receivable ?15000 <CR> ?0 <CR>
```

After entering a <CR>, the program logic has corrected entry no. 4 for you and will list on the screen the entry prior to the correction and the entry after the correction. This feature lets you view the entries in the "neighborhood" of the newly edited change. Don't forget to (R)ecord the final edited file or you will not have saved the edited value(s) to the current file. Experience has shown that it is normal to make keyboard mistakes when entering data, especially if you have a large amount of information to enter. The results of the editing of entry no. 4 are shown below:

```
Number      Acct        Account            Debit        Credit

  3          100      Escrow              2,209.50       0.00
  4          500      Accounts rec.      15,000.00       0.00
  5         1000      Fixed assets            0.00       0.00
```

As before, the program returns you to the system menu:

```
Select (A)dd,(C)hange,(D)elete,(E)nd,(I)nsert,(L)ist,(M)erge (N)o
       Assets/Liab,(P)rint,(R)ecord,(S)um debits/credits, (T)itle
       Accounts,(V)alues change,(X)company's name,
                                (Z)ero all values
                                ENTER your choice?
```

At this point, let's exercise the (P)rint option to check the work. Assuming you have a printer connected to a printer controller card in slot 1 of the Apple (the program was developed utilizing the Epson MX-80 or MX-100) enter the following:

```
P <CR>
```

The program will respond with the following sequence:

```
Align paper and READY printer...press <CR> when ready
```

This prompt directs you to check your printer for paper and paper alignment and to be sure the printer is on. When all is well, enter a <CR>.
The next prompt appears as:

```
INPUT identification? FIRST.BAL <CR>
```

This prompt asks you to provide a name or identification for your printout. We have selected "FIRST.BAL" as the name simply for convenience. Of course, any other identifier would work.

The next prompt is:

```
INPUT date ?6/5/83 <CR>
```

This input command will accept any date format you elect to enter such as "May 15,1983" or "5,May 83", etc. After pressing the <CR>, the program will print the results of FIRST.BAL on your line printer. The results of our current example should appear as shown in Table 4.2.

Printout Summary
File Name: B: FIRST.BAL

Identification: FIRST.BAL DATE: 5/15/83

| No. | Acct No. | Account Name | Debit Amount | Credit Amount |
|---|---|---|---|---|
| 1 | 1 | CHECKING ACCT | 1,000.00 | 0.00 |
| 2 | 2 | PETTY CASH | 120.00 | 0.00 |
| 3 | 100 | ESCROW | 2,209.50 | 0.00 |
| 4 | 500 | ACCTS REC. | 15,000.00 | 0.00 |
| 5 | 1000 | FIXED ASSETS | 0.00 | 0.00 |
| 6 | 1001 | COMPUTER EQUIP | 27,928.62 | 0.00 |
| 7 | 1002 | RESV FOR DEPR | 0.00 | 12,725.66 |
| | | | $46,258.12 | $12,725.66 |
| Sum of the credits less sum of the debits: | | | −33,532.46 | |

**TABLE 4.2: Printout of the Beginning FIRST.BAL File**

Examination of Table 4.2 reveals that our balance sheet is definitely not in balance. This was intended to be the case since we have purposely not provided all of the information required. We need to continue with additional data, and we will complete a series of transactions utilizing the (A)dd option. Also, we did not want to provide all of the data at one time since in reality you will be entering, updating, and editing information pertaining to your business periodically and not all at one time. The next section will review the remaining options of the system menu and present additional data that will result in a "balanced" balance sheet plus an income statement.

## 4.4 A Second Run to Input FIRST.BAL And FIRST.INC

Assume you have completed the input of the sample data and have obtained a printout of the results. All programs in this set were written under the assumption that you have an Epson MX-80 or MX-100 printer or a similar printer. You may have to modify the existing BASIC code for each of the programs if your particular printer is different from the Epson. In any event, a printer is a "must" for the accounting system to provide you with the necessary information in a hardcopy format.

At this point, we have produced a hard copy of our previously described input data, and we note that it is not in balance. Table 4.2 reveals that the sum of the credits less the sum of the debits is out of balance by -$33,532.46. Is something wrong? The answer is "no" because we have purposely designed this to occur. To illustrate our point, we will ask you to terminate the current run by entering an "E" followed by a <CR> to end the current run. Turn off the Apple and turn it back on again. Apple CP/M will come up with the "A>" prompt. Type MBASIC <CR> and wait for MBASIC to be active. Enter RUN "MENU <CR> and wait for the system menu to appear. Select the "S" option for (S)ETUP and respond with FIRST.BAL <CR> to the file name entered prompt. What we are doing is simply re-opening the file FIRST.BAL to show you that we can pick up where we left off. Our objective is to present additional ele-

ments of the example balance sheet by adding seventeen more entries to the existing seven entries we have already entered. At this point the following prompt is present:

```
DISK (A,B,C,D) ? B <CR>
```

The above prompt asks us to define the drive that contains our data disk (which is drive B).
Next, the system asks the following question:

```
Select input from (D)isk or (K)eyboard ?
```

Respond with a "**D**" followed by a <CR> to indicate that we want the file FIRST.BAL read into RAM from drive B. Within a few seconds the system menu will appear as:

```
Select (A)dd,(C)hange,(D)elete,(E)nd,(I)nsert,(L)ist,(M)erge (N)o
       Assets/Liab,(P)rint,(R)ecord,(S)um debits/credits, (T)itle
       Accounts,(V)alues change,(X)company's name,
                                (Z)ero all values
                               ENTER your choice?
```

Respond with an "**A**" followed by a <CR> to indicate that we wish to ADD information to the existing file FIRST.BAL. The system will automatically count the number of previous entries (seven in this case) and will display the following:

```
Entry No.  Acct.No.  Account Name   Debit Amt.   Credit Amt.
  8          ?
```

Note that the prompt character "?" is positioned under the column entitled "Acct. No.". We are now ready to begin adding additional data to the previous seven entries. Table 4.3 summarizes the remaining seventeen entries that are to be made to our hypothetical example.

| Number | Acct. No. | Account | Debit | Credit |
|---|---|---|---|---|
| 8 | 1030 | HOUSES | 180656.48 | 0.00 |
| 9 | 1031 | RES FOR DEPR | 0.00 | 21950.30 |
| 10 | 1400 | LOANS STOCKHOLDERS | 20937.32 | 0.00 |
| 11 | 1500 | MORTGAGES | 52627.30 | 0.00 |
| 12 | 1900 | LAND INVESTMENT | 59936.83 | 0.00 |
| 13 | 1901 | ORGANIZATION INV. | 21.10 | 0.00 |
| 14 | 2000 | LIABILITIES | 0.00 | 0.00 |
| 15 | 2001 | ACCTS. PAY. | 0.00 | 0.00 |
| 16 | 3999 | TAXES PAY. | 0.00 | 425.60 |
| 17 | 4000 | SHORT TERM NOTES | 0.00 | 0.00 |
| 18 | 4010 | DEPOSITS | 0.00 | 736.50 |
| 19 | 4020 | SECURITY DEPOSITS | 0.00 | 1042.36 |
| 20 | 4030 | SHORT TERM BUS LOAN | 0.00 | 15000.00 |
| 21 | 5000 | MORTGAGES | 0.00 | 282142.59 |
| 22 | 5900 | STOCKHOLD. EQUITY | 0.00 | 0.00 |
| 23 | 5998 | CAPITAL STOCK | 0.00 | 18200.00 |
| 24 | 5999 | SURPLUS | 0.00 | 8214.14 |

**TABLE 4.3: Remaining Entries for FIRST.BAL Example Problem**

Go ahead and input the entries as shown in Table 4.3 and terminate the data entry by entering a -1 for entry no. 25. Press <CR> and the program will write the updated file FIRST.BAL to the diskette in drive B. This process will take approximately 10 seconds. When the disk write operation is finished, the usual system prompt will appear as:

```
Select (A)dd,(C)hange,(D)elete,(E)nd,(I)nsert,(L)ist,(M)erge (N)o
       Assets/Liab,(P)rint,(R)ecord,(S)um debits/credits, (T)itle
       Accounts,(V)alues change,(X)company's name,
                                (Z)ero all values
                                ENTER your choice?
```

To review the entire file FIRST.BAL on the screen, respond with an "**L**" followed by a <CR>. After reviewing the screen, respond with a "**P**" <CR> to print the file FIRST.BAL on the printer. The results of this effort are summarized in Table 4.3. Note that the example problem data is in balance and all appears well.

**Printout Summary**
**File Name: B: FIRST.BAL**

**Identification: FIRST.BAL** **DATE: 5/15/83**

| No. | Account No. | Account Name | Debit Amount | Credit Amount |
|---|---|---|---|---|
| 1 | 1 | CHECKING ACCT | 1000.00 | 0.00 |
| 2 | 2 | PETTY CASH | 120.00 | 0.00 |
| 3 | 100 | ESCROW | 2209.50 | 0.00 |
| 4 | 500 | ACCTS REC. | 15000.00 | 0.00 |
| 5 | 1000 | FIXED ASSETS | 0.00 | 0.00 |
| 6 | 1001 | COMPUTER EQUIP | 27928.62 | 0.00 |
| 7 | 1002 | RESV FOR DEPR | 0.00 | 12725.66 |
| 8 | 1030 | HOUSES | 180656.48 | 0.00 |
| 9 | 1031 | RES FOR DEPR | 0.00 | 21950.30 |
| 10 | 1400 | LOANS STOCKHOLDERS | 20937.32 | 0.00 |
| 11 | 1500 | MORTGAGES | 52627.30 | 0.00 |
| 12 | 1900 | LAND INVESTMENT | 59936.83 | 0.00 |
| 13 | 1901 | ORGANIZATION INV. | 21.10 | 0.00 |
| 14 | 2000 | LIABILITIES | 0.00 | 0.00 |
| 15 | 2001 | ACCTS. PAY. | 0.00 | 0.00 |
| 16 | 3999 | TAXES PAY. | 0.00 | 425.60 |
| 17 | 4000 | SHORT TERM NOTES | 0.00 | 0.00 |
| 18 | 4010 | DEPOSITS | 0.00 | 736.50 |
| 19 | 4020 | SECURITY DEPOSITS | 0.00 | 1042.36 |
| 20 | 4030 | SHORT TERM BUS LOAN | 0.00 | 15000.00 |
| 21 | 5000 | MORTGAGES | 0.00 | 282142.59 |
| 22 | 5900 | STOCKHOLD. EQUITY | 0.00 | 0.00 |
| 23 | 5998 | CAPITAL STOCK | 0.00 | 18200.00 |
| 24 | 5999 | SURPLUS | 0.00 | 8214.14 |
| | | | $360,437.15 | $360,437.15 |

**Sum of the Credits less sum of the Debits:** −$0.00

**TABLE 4.4: FIRST.BAL Balance Sheet for the Example Data**

We now have our beginning period balance sheet from which we can initiate the construction of income and expense for a hypothetical month. When finished, we will have illustrated the power and flexibility of the setup program to establish the foundations necessary for you to keep accurate and timely records for your own business. The next section will review the construction of an income statement utilizing setup. As before, sample income and expense data will be provided for you to follow.

## 4.5 Completing the file FIRST.INC

The purpose of this section is to input information for the construction of an income statement using hypothetical data. Shown below are the initial statements and answers starting with naming the file and ending with the results shown in Figure 4.2, a printout of the income/expense statement.

As before, bring up the system with the (S)ETUP option and respond with the following:

```
Balance sheet or income/expense file name?FIRST.INC<CR>
Disk (A,B,C,D)?B <CR>
Maximum number of items in the Chart of Accounts?200<CR>
Select input from Disk or Keys (D/K)?K <CR>
```

| Number | Acct.No. | Account | Debit | Credit |
|---|---|---|---|---|
| 1 | ?Revenue? Redo from start | | | |
| 0 | | ?Revenue | ?0.00 | ?0.00 |
| 2 | ?6010 | ?Rent income | ?0.00 | ?1233.00 |
| 3 | ?6020 | ?Consulting income | ?0.00 | ?7500.77 |
| 4 | ?6030 | ?Interest income | ?0.00 | ?788.77 |
| 5 | ?6040 | ?Software sales | ?0.00 | ?756.33 |
| 6 | ?6050 | ?Capital gains | ?0.00 | ?8976.33 |
| 7 | ?6900 | ?Misc sales income | ?0.00 | ?456.22 |
| 8 | ?-1 | | | |

```
Select (A)dd,(C)hange,(D)elete,(E)nd,(I)nsert,(L)ist,(M)erge (N)o
       Assets/Liab,(P)rint,(R)ecord,(S)um debits/credits, (T)itle
       Accounts,(V)alues change,(X)company's name,
                                (Z)ero all values
                                ENTER your choice?
The -1 is again used to terminate the sequence.
```

Looking at the last series of input values, notice that we input "Revenue" as the account number for revenue. This is an error! The account number should have been 6000. Before making this correction, notice how your Apple reacts to the input of characters in place of a number. First, try to type the word "revenue" rather than its account number and notice that the computer will not accept this input. To correct the error in line 1 (number of the revenue account), answer the menu question with "C" or "c" for (C)hange: Note the following:

```
Number    Acct.          Account       Debit    Credit
 1        ?6000 ?Revenue     ?0         ?0
```

In this case, it was necessary to reenter the entire line. After completing the task and pressing <CR>, the computer will print:

```
Number        Acct        Account              Debit          Credit
  1           6000        Revenue               0.00            0.00
  2           6010        Rent income           0.00        1,233.00

Select (A)dd,(C)hange,(D)elete,(E)nd,(I)nsert,(L)ist,(M)erge (N)o
       Assets/Liab,(P)rint,(R)ecord,(S)um debits/credits, (T)itle
       Accounts,(V)alues change,(X)company's name,
                                (Z)ero all values
                                ENTER your choice?
```

Thus, the (C)hange option allows you to rectify any data entry mistakes after you have entered data. This example serves to illustrate the editing capabilities of (S)etup.

Table 4.5 contains the sample data for our example problem FIRST.INC. At this point, you should use Table 4.5 to build this file, list, and print it to assure yourself that you have entered the data correctly.

**FILE: FIRST. INC**

| No. | Acct. No. | Acct. Name | Debit | Credit |
|---|---|---|---|---|
| 1 | 6000 | Revenue | 0.00 | 0.00 |
| 2 | 6010 | Rent income | 0.00 | 1233.00 |
| 3 | 6020 | Consulting income | 0.00 | 7500.77 |
| 4 | 6030 | Interest income | 0.00 | 788.45 |
| 5 | 6040 | Software sales | 0.00 | 756.33 |
| 6 | 6050 | Capital gains | 0.00 | 8976.33 |
| 7 | 6900 | Misc. sales income | 0.00 | 456.22 |
| 8 | 9000 | Expenses | 0.00 | 0.00 |
| 9 | 9010 | Maintenance | 3405.55 | 0.00 |
| 10 | 9020 | Supplies | 1203.44 | 0.00 |
| 11 | 9030 | Depreciation | 11456.22 | 0.00 |
| 12 | 9400 | Interest | 7345.44 | 0.00 |
| 13 | 9500 | Operations | 1645.00 | 0.00 |
| 14 | 9510 | Utilities | 218.66 | 0.00 |
| 15 | 9520 | Insurance | 766.88 | 0.00 |
| 16 | 9530 | Real estate taxes | 566.77 | 0.00 |
| 17 | 9540 | Business taxes | 521.00 | 0.00 |
| 18 | 9550 | Other taxes | 122.00 | 0.00 |
| 19 | 9560 | Prof. services | 97.00 | 0.00 |
| 20 | 9600 | Travel | 123.00 | 0.00 |
| 21 | 9650 | Medical plan | 489.00 | 0.00 |

**TABLE 4.5: Example Data for FIRST.INC**

Assuming you have entered the entire data set from Table 4.5, go ahead and (L)ist the file to the screen, then (P)rint the file on the printer. If you accomplished everything correctly, your printout should appear as shown in Table 4.6.

Printout Summary
File Name: B: FIRST.INC

Identification: FIRST.INC DATE: 5/15/83

| No. | Acct. No. | Acct. Name | Debit | Credit |
|---|---|---|---|---|
| 1 | 6000 | Revenue | 0.00 | 0.00 |
| 2 | 6010 | Rent income | 0.00 | 1233.00 |
| 3 | 6020 | Consulting income | 0.00 | 7500.77 |
| 4 | 6030 | Interest income | 0.00 | 788.45 |
| 5 | 6040 | Software sales | 0.00 | 756.33 |
| 6 | 6050 | Capital gains | 0.00 | 8976.33 |
| 7 | 6900 | Misc. sales income | 0.00 | 456.22 |
| 8 | 9000 | Expenses | 0.00 | 0.00 |
| 9 | 9010 | Maintenance | 3405.55 | 0.00 |
| 10 | 9020 | Supplies | 1203.44 | 0.00 |
| 11 | 9030 | Depreciation | 11456.22 | 0.00 |
| 12 | 9400 | Interest | 7345.44 | 0.00 |
| 13 | 9500 | Operations | 1645.00 | 0.00 |
| 14 | 9510 | Utilities | 218.66 | 0.00 |
| 15 | 9520 | Insurance | 766.88 | 0.00 |
| 16 | 9530 | Real estate taxes | 566.77 | 0.00 |
| 17 | 9540 | Business taxes | 521.00 | 0.00 |
| 18 | 9550 | Other taxes | 122.00 | 0.00 |
| 19 | 9560 | Prof. services | 97.00 | 0.00 |
| 20 | 9600 | Travel | 123.00 | 0.00 |
| 21 | 9650 | Medical plan | 489.00 | 0.00 |
| | | | $27,959.96 | $19,711.10 |

Sum of the Credits less sum of the Debits: −$8,248.86

**TABLE 4.6: Printout of the Completed File FIRST.INC**

We suggest that you enter the above data, list and print the entered file. Compare the printed file to Table 4.6 line by line, then see if they agree. Assume that line 10 contains an incorrect entry. For example, the incorrect debit amount for supplies of 1230.44 was entered instead of 1203.44. To change the debit value in line 10, exercise the (V)alues change option. The sequence of prompts will be:

```
Select (A)dd,(C)hange,(D)elete,(E)nd,(I)nsert,(L)ist,(M)erge (N)o
       Assets/Liab,(P)rint,(R)ecord,(S)um debits/credits, (T)itle
       Accounts,(V)alues change,(X)company's name,
                                (Z)ero all values
                                ENTER your choice? V <CR>

INPUT line number to be changed? 10 <CR>

Entry No.    Acct. No.    Account Name      Debit Amt        Credit Amt
   10          9020         SUPPLIES     ?1203.44 <CR>    ?0.00<CR>
```

Following the <CR>, the program will list the 9th, 10th and 11th entries in the table for you to check. Below the screen listing will be the normal system menu. Don't forget to save the corrected file by using the (R)ecord option!

## 4.6 Some Additional Abilities

An examination of the program menu below:

```
Select (A)dd,(C)hange,(D)elete,(E)nd,(I)nsert,(L)ist,(M)erge (N)o
       Assets/Liab,(P)rint,(R)ecord,(S)um debits/credits, (T)itle
       Accounts,(V)alues change,(X)company's name,
                              (Z)ero all values
                              ENTER your choice?
```

shows that the following options have been discussed:

```
(A)dd (C)hange (E)nd (L)ist (P)rint (R)ecord (V)alue
```

The remaining options may be divided into two groups. First are those options that handle the chart of accounts being entered or updated. They include:

```
(D)elete (I)nsert (M)erge (S)um (Z)ero all values
```

The second group of options are used to input or update the numbers of the types of accounts, (N)umbers, and to input or update the name and address of your company, (X)company's name. Assume you have loaded FIRST.INC into the program and then listed the first ten items in the chart of accounts with the following results:

| Number | Acct | Account | Debit | Credit |
|---|---|---|---|---|
| 1 | 6000 | Revenue | 0.00 | 0.00 |
| 2 | 6010 | Rent income | 0.00 | 1,233.00 |
| 3 | 6020 | Consulting in | 0.00 | 7,500.77 |
| 4 | 6030 | Interest inc. | 0.00 | 788.45 |
| 5 | 6040 | Software sale | 0.00 | 756.33 |
| 6 | 6050 | Capital gains | 0.00 | 8,976.33 |
| 7 | 6900 | Misc sales in | 0.00 | 456.22 |
| 8 | 9000 | Expenses | 0.00 | 0.00 |
| 9 | 9010 | Maintenance | 3,405.55 | 0.00 |
| 10 | 9020 | Supplies | 1,203.44 | 0.00 |

```
Select (A)dd,(C)hange,(D)elete,(E)nd,(I)nsert,(L)ist,(M)erge (N)o
       Assets/Liab,(P)rint,(R)ecord,(S)um debits/credits, (T)itle
       Accounts,(V)alues change,(X)company's name,
                              (Z)ero all values
                              ENTER your choice?
```

Select the S option for sum and press <CR>. The results are:

```
Sum of the debits: 27959.97 Sum of the credits: 19711.1
Sum of the credits less sum of the debits:       -$8,248.86

Select (A)dd,(C)hange,(D)elete,(E)nd,(I)nsert,(L)ist,(M)erge (N)o
       Assets/Liab,(P)rint,(R)ecord,(S)um debits/credits, (T)itle
       Accounts,(V)alues change,(X)company's name,
                              (Z)ero all values
                              ENTER your choice?
```

You may be surprised to see that the sum of the debits is not equal to the sum of the credits. Remember, this file contains year-to-date earnings and expenses and is not a balance sheet. The difference between the debits and credits represents a profit if positive or a loss if negative. In this case, the sample company has an $8,248.86 loss for the year to date.

Now, type **Z** followed by a <CR.>.You will see the following:

```
This routine is for year ending zeroing
all income/expense to date information
YES IF YOU ARE SURE YOU WISH TO CONTINUE? YES <CR>
```

This is one of the few times you must give an exact answer to a question. In this case, if you do not answer **YES** in all upper case letters, the computer will not zero the file. The object is to make sure you do not zero a file in error. It is assumed you have answered positively and then returned to the program menu, selected the list option (lines 1,10) to see if the job was done, and pressed <CR>. You then see on the screen:

| Number | Acct | Account | Debit | Credit |
|---|---|---|---|---|
| 1 | 6000 | Revenue | 0.00 | 0.00 |
| 2 | 6010 | Rent Income | 0.00 | 0.00 |
| 3 | 6020 | Consulting In | 0.00 | 0.00 |
| 4 | 6030 | Interest Inc. | 0.00 | 0.00 |
| 5 | 6040 | Software Sale | 0.00 | 0.00 |
| 6 | 6050 | Capital Gains | 0.00 | 0.00 |
| 7 | 6900 | Misc Sales In | 0.00 | 0.00 |
| 8 | 9000 | Expenses | 0.00 | 0.00 |
| 9 | 9010 | Maintenance | 0.00 | 0.00 |
| 10 | 9020 | Supplies | 0.00 | 0.00 |

```
Select (A)dd,(C)hange,(D)elete,(E)nd,(I)nsert,(L)ist,(M)erge (N)o
       Assets/Liab,(P)rint,(R)ecord,(S)um debits/credits, (T)itle
       Accounts,(V)alues change,(X)company's name,
                              (Z)ero all values
                              ENTER your choice?
```

If you have performed the task of zeroing the values in error, you still have the option of selecting to end the run without recording the results. If you want to zero your income/expense statement, you now must record the new values, otherwise the old results will remain in the file.

Assuming that you ended the above activity without recording the zero values and have now reloaded FIRST.BAL into the program, select (M)erge from the program menu, and you will see:

```
File to be added ? FIRST.INC
```

Answer the question with the name of the income/expense account. The next question is:

```
Disk (A,B,C,D)? B <CR>
```

For users with more than two drives, the programs have a built in option of allowing the user to select the drive in which to put the disk with the file to be merged. Upon pressing <CR>, the file will be loaded behind the file already in the computer. You may now select the print option to see what you have. Table 4.7 summarizes the merged files **FIRST.BAL** and **FIRST.INC.**

**Printout Summary**
**File Name: B: FIRST BAL**

**Identification: MERGED FIRST.BAL & FIRST.INC** **DATE: 5/15/83**

| No. | Account No. | Account Name | Debit Amount | Credit Amount |
|---|---|---|---|---|
| 1 | 1 | CHECKING ACCT | 1000.00 | 0.00 |
| 2 | 2 | PETTY CASH | 120.00 | 0.00 |
| 3 | 100 | ESCROW | 2209.50 | 0.00 |
| 4 | 500 | ACCTS REC. | 15000.00 | 0.00 |
| 5 | 1000 | FIXED ASSETS | 0.00 | 0.00 |
| 6 | 1001 | COMPUTER EQUIP | 27928.62 | 0.00 |
| 7 | 1002 | RESV FOR DEPR | 0.00 | 12725.66 |
| 8 | 1030 | HOUSES | 180656.48 | 0.00 |
| 9 | 1031 | RES FOR DEPR | 0.00 | 21950.30 |
| 10 | 1400 | LOANS STOCKHOLDERS | 20937.32 | 0.00 |
| 11 | 1500 | MORTGAGES | 52627.30 | 0.00 |
| 12 | 1900 | LAND INVESTMENT | 59936.83 | 0.00 |
| 13 | 1901 | ORGANIZATION INV. | 21.10 | 0.00 |
| 14 | 2000 | LIABILITIES | 0.00 | 0.00 |
| 15 | 2001 | ACCTS. PAY. | 0.00 | 0.00 |
| 16 | 3999 | TAXES PAY. | 0.00 | 425.60 |
| 17 | 4000 | SHORT TERM NOTES | 0.00 | 0.00 |
| 18 | 4010 | DEPOSITS | 0.00 | 736.50 |
| 19 | 4020 | SECURITY DEPOSITS | 0.00 | 1042.36 |
| 20 | 4030 | SHORT TERM BUS LOAN | 0.00 | 15000.00 |
| 21 | 5000 | MORTGAGES | 0.00 | 282142.59 |
| 22 | 5900 | STOCKHOLD. EQUITY | 0.00 | 0.00 |

Printout Summary continued.

| | | | | |
|---|---|---|---|---|
| 23 | 5998 | CAPITAL STOCK | 0.00 | 18200.00 |
| 24 | 5999 | SURPLUS | 0.00 | 8214.14 |
| 25 | 6000 | REVENUE | 0.00 | 0.00 |
| 26 | 6010 | RENT INCOME | 0.00 | 1233.00 |
| 27 | 6020 | CONSULTING INCOME | 0.00 | 7500.77 |
| 28 | 6030 | INTEREST INCOME | 0.00 | 788.45 |
| 29 | 6040 | SOFTWARE SALES | 0.00 | 756.33 |
| 30 | 6050 | CAPITAL GAINS | 0.00 | 8976.33 |
| 31 | 6900 | MISC.SALES INCOME | 0.00 | 456.22 |
| 32 | 9000 | EXPENSES | 0.00 | 0.00 |
| 33 | 9010 | MAINTENANCE | 3405.55 | 0.00 |
| 34 | 9020 | SUPPLIES | 1203.44 | 0.00 |
| 35 | 9030 | DEPRECIATION | 11456.22 | 0.00 |
| 36 | 9400 | INTEREST | 7345.44 | 0.00 |
| 37 | 9500 | OPERATIONS | 1645.00 | 0.00 |
| 38 | 9510 | UTILITIES | 218.66 | 0.00 |
| 39 | 9520 | INSURANCE | 766.88 | 0.00 |
| 40 | 9530 | REAL ESTATE TAXES | 566.77 | 0.00 |
| 41 | 9540 | BUSINESS TAXES | 521.00 | 0.00 |
| 42 | 9550 | OTHER TAXES | 122.00 | 0.00 |
| 43 | 9560 | PROF. SERVICES | 97.00 | 0.00 |
| 44 | 9600 | TRAVEL | 123.00 | 0.00 |
| 45 | 9650 | MEDICAL PLAN | 489.00 | 0.00 |
| | | | $388,397.11 | $380,148.25 |

Sum of the Credits less sum of the Debits: −$8,248.86

TABLE 4.7: MERGED FIRST.BAL and FIRST.INC

The delete and insert options work in a very similar manner. You should first list the accounts in the neighborhood of the account to be deleted or the account to be added. After the program menu, the first question will be one of the following:

```
Number of inserted line 0 to default?
or
Number of deleted line 0 to default?
```

Deleting and inserting a line into your file can lead to some loss of data, and the option of cancelling the act was included just in case. Once you have indicated which line to insert or delete, there will be a delay, depending on where in the file the line is and how large the file is. For a file of 60 items and for the insertion of a line within the first 10, the delay may be a little lengthy. Since this is done so infrequently, the amount of time needed to do the task shouldn't be a problem.

The next section will review use of the options "T" (for Title Accounts), "X" (for company name) and "N" (for the Numbers option).

## 4.7 The Other Files

In the construction of balance sheets and income/expense reports, it is customary to have titles for each of the separate sections of the report. These "titles" are printed on each report and serve to provide "definition" for the report. In our accounting system, titles are provided by the "T" option in the system menu. Selecting the T option results in the following system prompt:

```
(N)ew file or (U)pdate ?
```

For our example problem, we are creating new titles; therefore, we will answer **N** for "NEW". This is a normal response for a first-time setup. If we needed to change an existing set of title accounts, we would have responded with a "**U** for "Update" of the current title set. Thus, the completed response will be:

```
(N)ew file or (U)pdate ? N
```

The next prompt will be:

```
Number of Title Accounts ? 4
```

This prompt asks for the number of title accounts the user wants for report generation purposes. For the example problem, we have selected 4 such account titles. They are:

2000 Liabilities
5900 Stockholders' equity
6000 Revenue
9000 Expenses

For our purposes, a title account is one that does not have values debited or credited to it but is used as the title for a part of the income/expense or balance sheet report.

When the following questions are asked, answer as indicated:

```
Input title account number 1 .?2000 <CR>
Input title account number 2 .?5900 <CR>
Input title account number 3 .?6000 <CR>
Input title account number 4 .?9000 <CR>
```

Notice that the questions are answered in order from low to high. After answering the last question you will see:

```
Number                                          Title number
  1                                                 2000
  2                                                 5900
  3                                                 6000
  4                                                 9000
99 to continue or number to correct? 99 <CR>
```

You have the option of correcting any value. If you wish to add or delete values, you will have to start over from the beginning. After you have typed 99 and pressed <CR>, you will see:

```
Information being saved in file named B:DATAF.INC
```

```
Select (A)dd,(C)hange,(D)elete,(E)nd,(I)nsert,(L)ist,(M)erge (N)o
       Assets/Liab,(P)rint,(R)ecord,(S)um debits/credits, (T)itle
       Accounts,(V)alues change,(X)company's name,
                                (Z)ero all values
                                ENTER your choice?
```

The last option in the program menu to be examined is X for company's name. Selecting this option results in the following:

```
(U)pdate or (N)ew name ? N <CR>
Company's name:? Internal Bytes Management, Inc. <CR>
Address? 123 Solid State Road <CR>
City? Electron <CR>
State? AL <CR>
Zip Code? lllll <CR>
Phone number? 555-lll-llll <CR>
```

The prompt **(U)pdate or (N)ew** asks whether or not you are creating a company address for the first time or you are updating the existing company name. For our example, we will answer with an **N** for "new." If we had been updating the existing company name, we would have responded with a **U** for "update." In this case, the program would have retrieved the existing company name from the current disk and allowed you to edit.

After completing the phone number prompt and pressing <CR>, you will see:

```
Number                                        Data
0                                  Internal Bytes Management, Inc.
1                                  123 Solid State Road
2                                  Electron
3                                  AL
4                                  lllll
5                                  555-lll-llll
99 to continue or number to change? 99

Select (A)dd,(C)hange,(D)elete,(E)nd,(I)nsert,(L)ist,(M)erge (N)o
       Assets/Liab,(P)rint,(R)ecord,(S)um debits/credits, (T)itle
       Accounts,(V)alues change,(X)company's name,
                                (Z)ero all values
                                ENTER your choice?
```

When you select this option again, you will go directly to the above listing and be able to change a single item without inputting the entire set of data.

## 4.8 Numbers

One option in the menu has not yet been examined. (N)umbers is the option that defines the account numbers that are assets, liabilities, stockholders' equity, income, and expenses. The program cannot be executed without this information. After typing **N**<CR>, you will see:

```
Input the highest account number for each of the following types
of accounts:

(N)ew series or (U)pdate?
```

For the example, the above question should be answered with **N** <CR> to indicate that this is a new series. The questions and answers that follow are necessary for the chart of accounts as they are designed.

```
Assets? 1901 <CR>
Liabilities? 5000 <CR>
Stockhol eq? 5999 <CR>
Income? 6900 <CR>
Expenses ? 9650 <CR>
```

The program continues by allowing you to make corrections:

```
Highest                                          Account Number
0                                                Assets
1                                                Liabilities
2                                                S.H. Equity
3                                                Income
4                                                Expenses
99 to continue or number to correct?
```

The logic behind our selection of accounts is based upon our discussion in Section 4.2. The user must adhere to the guidelines described in Section 4.2, as the accounting system has been designed around these general rules. It might be a good idea for you to review Section 4.2 again. For the example problem, the program will respond with:

```
                        Highest Account Number
0                           Assets                          1901
1                           Liabilities                     5000
2                           Stockhol eq                     5999
3                           Income                          6900
4                           Expenses                        9650
99 to continue or number to correct?_
```

After typing **99** <CR>, you will hear the disk drive operate saving your results.

This last option is very important. If you do not match your accounts correctly, the program will not know when to stop, and you will not obtain meaningful results. Be sure to exercise care in checking all of your data entries.

The last section of this chapter will review elements of the BASIC program that comprise the SETUP routine. If you have experience in MBASIC, you may want to examine the source code of the program and make customizing changes. If you do not have any experience with programming, you can omit this section for now.

## 4.9 Examining The Program

In the first line of the program, line 10 is the REMark statement with the words SAVE"chart.bas". All programs in this collection are identified in the same manner.

Line 20 has a CLEAR statement and a format for PRINT USING statements

later in the program. The CLEAR statement sets all variables to zero and reserves room for words, called string variables, in BASIC.

Line 30 is a DIMension statement that determines the number of title accounts you can have. If you expect to have less than 10 title accounts, you can remove this statement completely. This would increase your capacity a little. Seldom, if ever, would the number exceed 20 title accounts.

Lines 40, 50, and 60 clear the screen and then print the title of the program on the screen. Lines 70 through 100 generate the beginning of the file name input. Line 100 branches to a subroutine at line number 1730 where file name extents are validated. Lines 130 through 170 involve inputting the file name and the disk where the file is found. The disk, a colon, and the name of the file are combined in line 170 in the variable FILE$. This variable now may be used to OPEN a file for either input or output.

The program is designed to accept dimension size from the user. Line 180 is the line that performs this function. The first part of the line is the input statement, and the second part is the dimension statement for A$(.), A#(.,2), and A%(.). A$ is the name of the account, A# is a double-precision variable that constrains the debit and credit values associated with the account, and A% is the variable that contains the account number.

The next three lines, numbers 190,200 and 210, are designed to tell the computer where the information is coming from—disk files or the keyboard. The first time through, it must come from the keyboard, since no file exists. Notice how the LEFT$ pick out the leftmost character to obtain the match. This practice is followed throughout the program.

Lines 220 through 240 contain the routine that reads the chart of accounts file. Notice that the program counts the number of lines in the file as the file is input. The input is stopped when the program finds the word "END" in the name position in the file. There are 2 methods for determining the number of items in the file. The first number read in, called DUMMY in line 220, contains this information and the trailing entry should have the word "END" recorded. If you are planning to change the program in any way, make sure you maintain both the beginning information and the end-of-file marker, since both are used in the programs. Line 250 counts the total number of records in the file.

Lines 260 and 270 are a short subroutine that prints a heading used at various times during program operation.

Lines 280, 290, and 300 are a routine for inputting information into the chart of accounts file. This routine may be entered in line 280 or line 290 depending upon whether you are updating or starting the file. This is the reason Z=1 is the only thing in line 280.

Lines 310 through 340 are designed to list the chart of accounts file. This routine will be entered in line 310 for the listing option in the program menu and in line 320 for other listing tasks associated with different program menu options.

Lines 350 through 400 establish a FOR-NEXT loop to list the output according to the PRINT-USING formats embedded in the loop.

Line 410 sets up the program menu defined in the subroutine located at lines 1820 through 1870. Lines 420 through 570 are the IF statements that route

the program depending on the selection made in the menu. All the IF statements use the LEFT$ function to find the first character input and are designed to react to both upper- and lowercase answers.

Lines 580 through 610 contain the change option. Line 580 inputs the actual line number to be changed. Line 590 goes to a subroutine to print headings, and line 600 performs the act of inputting the new values. Line 610 then sets the upper and lower values to be listed and goes to the listing routine to list the lines.

Lines 620 through 700 contain the record option. The OPEN statement which prepares the system to write information to the diskette is in line 620. In line 700 is a CLOSE statement that tells the computer the task of writing to the diskette is complete. It is good practice to leave a file open for the minimum amount of time. As soon as the task of inputting or outputting to a file is complete, the file is closed. An open file can be damaged by a power surge, and this practice minimizes this source of trouble.

A key problem in all accounting programs is controlling the rounding that occurs from the fraction of pennies often produced by the manner in which the computer handles numbers. In lines 640 through 660 is a rounding correction routine that is run each time the computer is asked to record the values with which it is working.

The insert routine is contained in lines 710 through 780. The number of the line to be inserted is input in line 710. The subroutine prints the headings, and lines 740 through 760 move the data around to allow for the data entry. This routine takes time. Line 770 does the actual inputting, while line 780 directs the flow of the program to the task of listing the file in the neighborhood of the inserted line.

Lines 790 through 860 are the delete routine. It is organized similar to the insert routine. First, the line to be deleted is identified, then the file is moved up to replace the line. The word "END" then is placed in the last account. Line 800 reduces the number of accounts counter Z.

Lines 870 through 910 are written as a subroutine for the purpose of summing the debits and credits. The value of DEBIT# and CREDIT# are first set to zero, and then the addition starts. All values are maintained as double-precision values to maximize accuracy.

Lines 920 through 1110 contain the routine designed to print the information in the files on the printer. The print formats have been designed for 80 column printer capability. If, for some reason, you desire to alter or reformat the print output, your individual modifications will occur in line 920 through 1110. The LPRINT statement contained in line 1100 is a forms feed command recognized by the Epson printer.

The task of merging a second file to the information already in the computer is performed by lines 1120 through 1150 from lines 230, 240, and 250. Again, notice that we input the number of items in the file and then used the trailing account method to find out when the end of the file is reached.

Changing the information about a specific account is accomplished in lines 1160 through 1200. In line 1200 the computer is directed to list some of the lines in the neighborhood of the change.

The act of zeroing the values in the income/expense file to prepare to start a new year occurs in lines 1210 through 1220. In line 1220, if the exact answer is not **YES** in upper- case letters, the flow of the program is directed to the program menu and not allowed to continue.

The tasks of inputting and updating the information about the title accounts are done in lines 1250 through 1410. Notice that each time a file is opened for either input or output, it is closed as soon as the task is complete. Notice also that the first value in the file is the number of items in the file. NOTITLE is the name given to this variable.

The number option is designed to tell the computer which accounts are assets, which are liabilities, etc. This information is controlled by the statements in lines 1420 through 1550. Both the input of new file information and update of information contained in an old file is performed by these lines.

The last routine inputs and controls the information about your company's name and address. The code to perform this task is written in lines 1560 through 1720.

## 4.10 Summary

The detailed operation of (S)ETUP has been reviewed by going through an example problem. The example is intended to represent a hypothetical case and the creation of the files FIRST.BAL and FIRST.INC represents beginning-of-period accounting information. If you key in the steps yourself, you will be able to walk through the example program included to help you find possible keypunch errors. If you have purchased the coding complete, it is still important to walk through the example given to make sure you understand all the instructions as the program is used.

(S)ETUP must be used to set up your chart of accounts when the system is first used. It also must be used to update and change the chart of accounts to meet the ever changing needs of an operating business. Be sure to backup the data disks and keep them out of the wrong hands.

The next chapter will cover the entry of periodic data such as daily, weekly, or monthly transactions that occur as a result of the cash flow or credit operations of the business. In this chapter, you will be exposed to the computer "posting" of transactions to the chart of accounts that have been created by the (S)ETUP program. Thus, it is important that you have carefully reviewed the procedures outlined in this chapter in order for the (T)RANSACTION program to function normally.

# 5. (T)ransaction Program

The organization of the (T)ransaction program is very similar to the Setup program. These routines have similar functions, such as to input, correct, update and they have a file of variables that are sometimes integer, sometimes characters, and sometimes double-precision numbers. An effort was made to make these programs as similar as possible, since this minimizes what the user has to learn. Despite this similarity, we will review in detail all options of this program menu and the program code. We also will give a few hints on how the program may be used efficiently to minimize the effort of inputting periodic transactions.

The number of transactions that may be handled in a single run is around 300 for a 56K Apple CP/M configured system. The period associated with a given run may be a day, a week, or any other period. Most businesses using this system run monthly transaction programs. However, you may require daily or weekly runs to record your transactions. Regardless of the frequency of data entry, it is suggested that you set aside a period of time on a regular basis to record transactions and stay with it until it becomes a "habit."

The task of running a general ledger system requires the use of this program on a periodic basis, usually monthly. Every period, the transactions for that period must be entered into the computer using this program. The program name reflects the information to be entered, i.e., the individual transactions.

A transaction is writing a check, making a bank deposit, issuing an accounts payable invoice, or paying a mortgage. The type of transaction you encounter is a function of the type of business you are in. The act of recording, or "posting," a transaction is similar for all businesses. The cash account is a debit balance (lefthand column) when it is positive. Receiving cash starts with debiting the cash account. Paying a bill starts with crediting the cash account to reduce its value. All other aspects of a transaction are built on the above fact.

For example, a mortgage payment starts with the issuing of a check. The amount of the check is a credit, since the issuing of a check means the cash account must be reduced. To balance this entry, interest expense, reduction of loan, and escrow must all be debit entries. The sum of the debits and the sum of the credits must balance at the end of each transaction.

## 5.1 Starting With Reusable Files

Transactions such as depreciation expense do not change from period to period within a given year. It is recommended that you set up a special file for such transactions and then merge this file with your other transaction information to minimize typing and potential errors. The specific transaction that may be kept in such a file depends on your particular business.

The (T)ransaction program is called from the menu. This program starts with the following screen prompt:

```
TRANSACTION ROUTINE
by
Steven M. Zimmerman, Ph.D.
Leo M. Conrad
Donald R. Smith Ph.D., P.E
*** (press <CR> to begin) ***?
```

**Figure 5.1: Initial Prompt Screen for Transaction**

After pressing <CR>, the program clears the screen and produces the following prompt:

```
Transaction Program
```

* PURPOSE: To post or record periodic transactions to a data file. It is suggested that the name of the file correspond to the month the transactions occur or describe the nature of the file.

```
EXAMPLE FILE NAMES: MAY83, DEPR, LOANS,... etc.

ENTER Transaction file name ?
```

This prompt asks for the file name under which the transactions that are to be entered will be saved. Use the name of the time period from which the transactions come except when you are setting up a file to be used again for different periods, such as a DEPreciation file. Remember, there is a seven character limit to the file name.

The next prompt is:

```
Disk (A,B,C,D)?
```

As in Setup, this prompt asks for the disk drive that your data file diskette is in, usually drive B. We recommend that you use drive A for programs and drive B for data. If you are an accounting firm using this software, you could use the same diskette A for all clients and then store the transactions for each individual client on a separate diskette. A business with a chart of accounts of some 50 items and 100 transactions per month should fit on a single-sided Apple CP/M diskette. If you have a system with double-sided disk drives, the size of the business could be doubled.

After entering the disk drive, the next prompt is:

```
DATE?
```

This prompt asks for the current date. You can enter the date in any form you choose, i.e., 5/15/83 or May 15, 1983, etc. The next prompt is:

```
Maximum number of transactions?
```

This prompt is asking for the "maximum" number of transactions you expect for this particular file. After entering this number, the next prompt is:

```
Select input from (D)isk or (K)eys?
```

As in the Setup program, this prompt is asking for which mode of data entry is being used. If you are creating a first-time transaction file, then you will select the (K) option for keyboard entry. If you want to edit, review, list, merge or print an existing file, you will select the (D) option for Disk file input.

At this point, we will create a short example involved with the creation of a depreciation file for computer equipment and residential property. We will use the file name DEP, and this will be a first-time entry. Therefore, we will use the (K)ey option for data entry. After we have finished inputting the data for DEP, we will save the file for later use with other transactions. When we are finished, we will have a single transaction file which we will use in Chapter 6 to produce a balance sheet and an income/expense report.

To begin, place a copy of the accounting system in drive A and an Apple CP/M formatted diskette in drive B. Turn on the Apple and, when the CP/M prompt appears, perform the following:

```
>A MBASIC <CR>
```

or,

```
>A MBASIC MENU <CR>
```

When the MBASIC prompt appears, enter

```
RUN ''MENU <CR>
```

from the keyboard. When the menu appears, select the **T** option for the TRANSaction program. The menu program will load the TRANSaction program and the main promotion prompt will appear. After pressing <CR>, respond to the prompts as indicated:

```
ENTER Transaction File name? DEP <CR>
Disk (A,B,C,D)? B
DATE:? 5/15/83 <CR>
Maximum number of transactions:? 25 <CR>
Select input from (D)isk or (K)eys ? K <CR>
```

At this point, the (T)ransaction program will respond with a screen layout as shown in Table 5.1.

| Trans No ? | Date | DESCRIPTION | ACCOUNT DEBIT | ACCOUNT CREDIT | AMOUNT DEBIT | AMOUNT CREDIT |
|---|---|---|---|---|---|---|

**TABLE 5.1: Input Format for Transaction Program**

The purpose of this screen format is to set up a columnar entry scheme to assist in data entry. An explanation of each column follows:

Trans- a user-defined transaction number assigned at the time of data entry. The number may be a check number or any number you, the

user, wishes to assign to the transaction. For example, we suggest using "999" for transactions such as posting depreciation amounts. Such transactions do not have checks or vouchers associated with them and are periodically posted to the accounts to reflect their "existence." For transactions that do have checks or some other written record, we suggest using some appropriate transaction numbering scheme.

Date- This column is for entering the date of the transaction in a compressed form such as "5/1", for May 5, 19xx. If you do not desire to associate a date with the transaction, enter "999".

DESCRIPTION- This column is for entering a short description of the transaction such as "Dep. Comp. Equip.". Try to keep the description as short as possible yet long enough to accurately communicate the nature of the transaction.

ACCOUNT
DEBIT CREDIT

This column is comprised of two sub-columns. The major column is to identify the account number that is to be debited and the account number to be credited. Account numbers entered in these two columns must satisfy the requirements of valid account numbering described in Chapter 4.

AMOUNT
DEBIT CREDIT

This column is similar to the previous column except that the respective amounts for debiting and crediting are entered.

To illustrate the data entry, assume we have two depreciation amounts we wish to post to a file called DEP. One entry is for the periodic depreciation of computer equipment (account 9030) for the sum of $213.43. The second entry is for depreciation associated with rent property (account 9030) amounting to $542.19. The entries under the screen input format are shown in Table 5.2.

| Trans No | Date | DESCRIPTION | ACCOUNT DEBIT | ACCOUNT CREDIT | AMOUNT DEBIT | AMOUNT CREDIT |
|---|---|---|---|---|---|---|
| ?999 | ?999 | ?Dep Computer Equip | ?9030 | ?1002 | ?213.43 | ?213.43 |
| ?999 | ?999 | ?Dep Houses | ?9030 | ?1031 | ?542.19 | ?542.19 |
| ? -1 | | | | | | |

**TABLE 5.2: Example Responses for the DEP File**

The input sequence is stopped in the same manner as in the setup program. Remember: Using a -1 is easier when it is used consistently in both programs. After inputting -1 and pressing <CR>, you will see the Transaction program menu:

```
(A)dd,(B)alance,(C)hange,(D)elete,(E)nd,(I)nsert,(K)lear,(L)ist
  (M)erge,(P)rint,(R)ecord,(S)um debits/credits,(V)alues
                    ENTER your choice?
```

You may answer the menu with a complete name in upper- or lower case or simply the first letter of the option of interest. If you make a selection not listed in the menu you will be asked to make another selection. We are going to start with **L** for the listing option. After making this selection and pressing <CR>, you will see:

```
INPUT THE LINE NUMBERS TO BE LISTED
STARTING LINE NO.                                    ENDING LINE NUMBER
     ?1 <CR>                                              ?4 <CR>
```

By responding with a **1** and a **4**, we are requesting a listing of lines 1 through 4 of the transactions just entered. This is exactly the same question you were asked when you selected the listing option in the SETUP program. The results are shown in Table 5.3.

| No Trans No | Date | DESCRIPTION | ACCOUNT DEBIT | ACCOUNT CREDIT | AMOUNT DEBIT | AMOUNT CREDIT | St |
|---|---|---|---|---|---|---|---|
| 1 999 | 999 | Dep. compute | 9030 | 1002 | 213.43 | 213.43 | IS |
| 2 999 | 999 | Dep. houses | 9030 | 1031 | 542.19 | 542.19 | IS |
| 3 -1 | | | 0 | 0 | 0.00 | 0.00 | |
| 4 0 | | | 0 | 0 | 0.00 | 0.00 | |

**TABLE 5.3: Example screen listing of the DEP file**

At the conclusion of the screen listing, the Transaction menu will appear as:

```
(A)dd,(B)alance,(C)hange,(D)elete,(E)nd,(I)nsert,(K)lear,(L)ist
  (M)erge,(P)rint,(R)ecord,(S)um debits/credits,(V)alues
                        ENTER your choice?
```

Look at the last item listed in the above table. The letters IS for ISSUED may be seen. This information is included to aid in balancing the value in your cash account with the bank's value. The transactions in the DEP file should be classified as CLEARED rather than ISSUED. To do this the (K)lear option is used. (The letter K is used because the letter C already was used in the menu.) Selecting the K option results in the following question:

```
Input transaction starting number? 1
```

The computer now walks you through the file starting with the transaction number indicated:

```
Trans Date DESCRIPTION          ACCOUNT        AMOUNT        St
No                              DEBIT   CREDIT DEBIT  CREDIT
999   999  Dep. Computer equip 9030    1002   213.43 213.43 ISSUED
(A)justment, (C)leared or (I)ssued, (E)nd? C <CR>
999   999  Dep. Houses         9030    1031   542.19 542.19 ISSUED
(A)justment, (C)leared or (I)ssued, (E)nd? C <CR>
```

All ISSUED have been changed to CLEARED. Since you have reached the end of the file, the computer will automatically return to the program menu.

At this point, select the **R** option and record the file. After the file is recorded, select the **E** option to end the session. We now have a completed file called DEP that has been recorded on drive B. In the next section, we will create another file called MAY for the purpose of posting other transactions in addition to the ones already recorded in DEP. We will merge the DEP file to the MAY file and record the merged file which will be used in producing financial reports covered in Chapter 6.

## 5.2 A Monthly Run

After entering **E** to End the DEP transactions, Type RUN to start the program over again. This time, when asked for the name of the transaction file, answer MAY as shown below: Transaction file name? **MAY**

A month's name was used to illustrate how a file name should be tied into a time period. The maximum number of transactions question may be answered with a 300 (if you have a 64 KB system), and the select input question with a K. As soon as the transaction input is asked for, type **-1** to terminate the task. You now should see the program menu:

```
(A)dd,(B)alance,(C)hange,(D)elete,(E)nd,(I)nsert,(K)lear,(L)ist
  (M)erge,(P)rint,(R)ecord,(S)um debits/credits,(V)alues
                        ENTER your choice?
```

This time, select the (M)erge option in the menu. You then will see the following on your monitor:

```
File to be added? DEP <CR>
```

The idea is to bring the DEP transactions into a file named MAY. If we do not merge the DEP file with MAY, the transactions recorded in DEP will not be included in any of the financial reports. Therefore, it is extremely important to remember to merge all secondary transactions (like DEP) into the main transaction file (MAY) before running the Report program (discussed in Chapter 6).

The next question is:

```
Disk (A,B,C,D)? B
```

This completes the (M)erge questions. You will hear the disk drives work and will be returned to the program menu.

```
(A)dd,(B)alance,(C)hange,(D)elete,(E)nd,(I)nsert,(K)lear,(L)ist
  (M)erge,(P)rint,(R)ecord,(S)um debits/credits,(V)alues
                        ENTER your choice?
```

You are now ready to start adding transactions to the depreciation transactions to complete the month's file. Select the (**A**)dd option and input the transactions as shown in Table 5.4.

| Trans No | Date | DESCRIPTION | ACCOUNT DEBIT | ACCOUNT CREDIT | AMOUNT DEBIT | AMOUNT CREDIT |
|---|---|---|---|---|---|---|
| ?999 | ?5/2 | ?Rent apartments | ?1 | ?6010 | ?1500 | ?1500 |
| ?999 | ?5/3 | ?Consulting | ?1 | ?6020 | ?2200 | ?2200 |
| ?135 | ?5/3 | ?Interest business loan | ?9400 | ?1 | ?125 | ?125 |
| ?136 | ?5/3 | ?Mortgage payment | ?9400 | ?1 | ?133 | ?300 |
| ?136 | ?5/3 | ?...reduction mor | ?5000 | ?1 | ?167 | ?0 |
| ?-1 | <CR> | | | | | |

**TABLE 5.4: Example Entries for MAY Transaction File**

After pressing <CR> you will again see the program menu:

```
(A)dd,(B)alance,(C)hange,(D)elete,(E)nd,(I)nsert,(K)lear,(L)ist
  (M)erge,(P)rint,(R)ecord,(S)um debits/credits,(V)alues
                    ENTER your choice?
```

Of interest in the above set of transactions is how the compound entry needed for the mortgage is handled. Two lines of input were needed for this entry. The first line inputs the interest expense and the reduction in the cash account, while the second line inputs the reduction of the mortgage and indicates no change in the cash account. The design concept of this program is based on the traditional double entry accounting idea that debits may be used as a check against the credits to reduce the number of errors.

Assume your inspection of the above listing indicated that there was an error in the amount of dollars typed for "rent apartments". The value shown above is $1,500, and the correct value is $1,900. To correct this error, you may use either the change option or the value option. It is easier to use the value option, since all that must be changed are the accounts debited and credited and the amounts debited and credited. Selecting **V** results in the following prompts and responses:

```
INPUT line number to be changed? 3 <CR>
```

The program responds on the screen with:

```
No                    DESCRIPTION      ACCOUNT      AMOUNT      St
Trans      Date
 No                              DEBIT  CREDIT  DEBIT  CREDIT
999   5/2   Rent Apartments       ?1     ?6010   ?1900  ?1900   ?ISSUED
After entering a <CR>, the screen will display the following:
```

```
No
Trans Date    DESCRIPTION    ACCOUNT            AMOUNT              St
No                            DEBIT   CREDIT    DEBIT     CREDIT
  3    5/2  Rent apartmen      1      6010    1,900.00  1,900.00  IS
  4    5/3  Consulting         1      6020    2,200.00  2,200.00  IS

(A)dd,(B)alance,(C)hange,(D)elete,(E)nd,(I)nsert,(K)lear,(L)ist
  (M)erge,(P)rint,(R)ecord,(S)um debits/credits,(V)alues
                        ENTER your choice?
```

The next selection is the ability to insert a line. This starts with typing **I** as the answer to the above menu. The questions you will see and our sample answers are:

```
Number of inserted line 0 to default? 5 <CR>
```

The screen will then display:

```
No
Trans   Date   DESCRIPTION    ACCOUNT            AMOUNT
No                            DEBIT     CREDIT   DEBIT      CREDIT
?134    ?5/3   ?Maintenance   ?9010     ?1       ?150      ?150 <CR>
```

After pressing <CR>, the screen display will be:

```
No
Trans
No      Date   DESCRIPTION      ACCOUNT        AMOUNT                St
                              DEBIT   CREDIT    DEBIT      CREDIT
4 999   5/3   Consulting       1       6020    2,200.00   2,200.00   IS
5 134   5/3   Maintenance      9010    1         150.00     150.00   IS
6 135   5/3   Interest bus     9400    1         125.00     125.00   IS

(A)dd,(B)alance,(C)hange,(D)elete,(E)nd,(I)nsert,(K)lear,(L)ist
  (M)erge,(P)rint,(R)ecord,(S)um debits/credits,(V)alues
                        ENTER your choice?
```

The word Maintenance was spelled with an "x" rather than "a" so we could use the change option. When you type **C** in answer to the above menu, you will see:

```
INPUT line number to be changed? 5 <CR>
```

The screen will display:

```
No                                ACCOUNT          AMOUNT
Trans   Date   DESCRIPTION                                        St
No                            DEBIT   CREDIT   DEBIT   CREDIT
?134    ?5/3   ?Maintenance   ?9010   ?1       ?150    ?150     ISSUED
```

The program will continue by listing the change and some lines in the neighborhood of the change.

To delete a line, start with the program menu and select the **D** option. You will be asked:

```
Number of deleted line 0 to default?
```

As in the case of the insert option, you can stop this option by selecting 0 at this point. This is very important, because once the line is deleted, unless

you have recorded the results, it is lost. If you wish, select **0** to return to the program menu, record the current file, and then select D again to complete this option. You may now select any number and delete the line.

The (S)um option adds your debits and credits. It is an audit trail balance that may be run at any time. The sum of the debits and credits must be equal before you can produce an income/expense statement and a balance sheet.

We are now ready to examine the (P)rint option. At this point, we have entered the file MAY and (M)erged the DEP file with it. We will now select the (P)rint option to provide us with a listing of the file MAY. From the system menu:

```
Select (A)dd,(C)hange,(D)elete,(E)nd,(I)nsert,(L)ist,(M)erge
   (N)o Assets/Liab,(P)rint,(R)ecord,(S)um debits/credits,
   (T)itle Accounts,(V)alues change,(X)company's name,
               (Z)ero all values
               ENTER your choice?
```

Select the (P)rint option by entering a **P** followed by a <CR>. The first question you will be asked is:

```
INPUT identification of run? MERGED MAY AND DEP FILE <CR>
```

You may add any title you wish to the run. We seldom use this option, as the date and file name are printed automatically. However, we have selected the phrase "MERGED MAY AND DEP FILE" for illustrative purposes.

The next question is:

```
Printer menu
  (A)ll transactions
  (O)ne account? A <CR>
```

On occasion, when we wish to examine the activity in one particular account, the (O)ne option allows us to do so. Most of the time we use the (A)ll option to get a complete listing of the transactions for a period. The act of printing information concerning a specific account number is similar to the maintenance of a specialized journal. Manual accounting systems are often designed around sets of special accounts called "journals." This is not necessary when it is possible to create one of these special sets of books on a moment's notice. For our purpose, we have elected to use the (A)ll option.

The next questions follow:

```
Cash account #? 1 <CR>
Amount of original balance? 12500 <CR>
```

The printer prints a running balance for the account you identify as the cash account. The amount of the original balance is the dollars from the last balance sheet in the account in question. For our example problem, the cash account is account no. 1 and the balance of this account is $12,500.00. Table 5.5 summarizes the results of (P)rinting the file MAY.

***IDENTIFICATION: MERGED FILES..MAY AND DEP FILE: B:MAY
Date: 6/25/83

Original Balance: $12,500.00

| Trans No | Date | DESCRIP | ACCOUNT DEBIT | ACCOUNT CREDIT | AMOUNT DEBIT | AMOUNT CREDIT | S | |
|---|---|---|---|---|---|---|---|---|
| 1 999 | 5/2 | RENT APART | 1 | 6010 | 1,900.00 | 1,900.00 | IS | 14,400.00 |
| 2 999 | 5/3 | CONSULT INC | 1 | 6020 | 2,200.00 | 2,200.00 | IS | 16,600.00 |
| 3 134 | 5/3 | MAINTENANCE | 9010 | 1 | 150.00 | 150.00 | IS | 16,450.00 |
| 4 135 | 5/3 | INTEREST | 9400 | 1 | 125.00 | 125.00 | IS | 16,325.00 |
| 5 136 | 5/3 | MORTGAGE | 9400 | 1 | 133.00 | 300.00 | IS | 16,025.00 |
| 6 136 | 5/3 | ...REDUCTION | 5000 | 1 | 167.00 | 0.00 | IS | 16,025.00 |
| 7 999 | 999 | COMPUTER DEP | 9030 | 1002 | 213.43 | 213.43 | CL | 16,025.00 |
| 8 999 | 999 | DEP. HOUSES | 9030 | 1031 | 542.19 | 542.19 | CL | 16,025.00 |
| | | | | | 5,430.62 | 5,430.62 | | |

**TABLE 5.5: Example printout of the file MAY**

Exercising the (P)rint option allows you to review the periodic transactions for errors and/or omissions using the source documents to check against the computer printout. The user is encouraged to use this option frequently as it assists in catching errors that can cause problems.

The only option in the program menu we have not studied is (B)alance. This option will not be examined at this time. Its only application is in balancing the cash account. This topic will be covered in Chapter 9. As such, the Transaction program is used more than one time within the framework of our system. After reviewing Chapter 9, you will soon master the applications of the Transaction program.

## 5.3 Examining The Program

The Transaction program is structured in a manner similar to the Setup program. It starts with the initial input and is built around the program menu. A brief description of the major program elements follows.

Lines 10 through 50 name and identify the program, clear room for string variables, and specify some of the formats for our print-using statements. Line 30 then clears the screen and prints the title line on the screen.

Line 60 references a subroutine at line 1660 that provides the initial screen "promo." Lines 70 and 80 ask the first prompt concerning the name of the file and disk drive to be used. Lines 90 through 100 form an "error trap" that aids in checking for an invalid disk drive designation. Line 110 combines the disk drive specification with the file name to allow us to open the file when needed.

Line 120 prompts for the data and line 80 continues by inputting the maximum number of transactions to be allowed in our file and then sets the dimension of the transaction number, account debited, account credited

(A%(i,3)), the date, description, issued (A$(i,3)), the amount debited, and the amount credited (A#(i,2)).

The option of starting from the keyboard or from a disk file is selected in line 140, and the subroutine that does the task is selected in line 150. If an illegal option was selected in line 140, the program returns the user to line 140 in line 160.

Lines 170 through 200 input the information from a disk file. When complete, the program moves to the program menu in line 370. Note: The word "END" in the date position in line 190 is used to terminate the input activity. This trailing information is needed in the file.

Contained in lines 210 through 260 is a subroutine that prints the headings on your screen. These are the headings found in the sample run in this chapter.

Lines 270 through 300 are used to put the initial information in the file. By starting in line 280, the code also is used to add information to the file.

Lines 310 through 360 are used to list the file on the screen. In the case of the listing option, line 310 is used.

The program menu is found in lines 370 through 390. Whenever an option is complete (with the exception of the end option), the program is returned to this line. The routine to select the option is performed in lines 400 through 520. Note: Line 530 returns the user to line 370 when an unacceptable selection has been made.

Lines 540 through 570 are designed to input a complete change of the information in a particular line of the data file. When the information is complete, the program automatically goes to the listing lines and lists the data file in the neighborhood of the change.

The data file is recorded in lines 580 through 660. Note: The file is opened (line 580) at the beginning of this task and then closed again as soon as the file is complete. Included in this set of lines is the task of correcting the data for rounding errors and printing the trailing data line with an END in all string variable positions and a zero in all value positions. The number of items in the file is not printed at the beginning of the file.

The insert routine is found in lines 670 through 750. The task of moving all data in the file to make room for the new line of data is what makes this program slow. This task is performed in lines 690 through 720. When the routine is complete, it goes to line 210 to list the data lines in the neighborhood of the inserted line.

The deletion routine is similar to the insertion routine. This task starts in line 760 and continues through line 830. The time needed to perform this activity is dependent on the number of data lines following the one deleted.

The summing of debits and credits is performed by the subroutine in lines 840 through 870. This routine prints the results as well as performing the addition.

The print routine is in lines 880 through 1140. As described in the Setup program in Chapter 4, you may wish to customize this section of the code to take advantage of any special printing features your printer may possess. Included within this routine is the adding of debits and credits, calculation of a

running balance for the account identified as the cash account, and printing the transactions in the data file.

The task of merging a file is performed in lines 1150 through 1180. The file to be merged, name, and disk, is input in line 1150. The name and disk information is combined to form the file name in line 1160 and the file is opened for input in line 1330.

The routine that clears the transaction starts in line 1190. Three alternate transaction codes are used: KLEAR, ADJUST, and ISSUED. KlEAR and ISSUED were reviewed briefly and ADJUST will be discussed when the checkbook balancing task is covered. The routine terminates in line 1330.

Lines 1430 through 1660 are designed to handle the cash balancing routine. The routine adds when the cash account is found as a credit and the ISSUED code is seen. It subtracts when the cash account is found as a debit and the ISsued code is seen. When the ADJUSTED code is found, the routine adds for debit cash and subtracts for credit cash. Remember, this is an adjusting routine. ISSUED means the transaction has been recorded but has not cleared the bank. ADJUST means the transaction has not been recorded but has cleared the bank!

## 5.4 Summary

This transaction routine produces a corrected set of transactions for a given time period. This information, combined with the old balance sheet information and the old income/expense information to date, combined with the files that contain the firm's name and other such information, allows for the production of an income/expense report and a balance sheet. The next chapter will discuss the utilization of the transaction files to produce meaningful financial statements.

This chapter has reviewed the Transaction program from a user's and then from a programmer's view. You must understand how the program works to use it or to rewrite it. You do not have to understand the code if all you are interested in is using the program.

# 6. REPORT: Income/Expense and Balance Sheet

This chapter describes a program called REPORT that combines the current TRANSACTION file with the "old" income/expense file, the "old" balance sheet file, the asset number file and the title file to produce a printed income/expense statement and balance sheet for the current period and a "new" or current income/expense and balance sheet file to be used for the next period. This program requires up to four files to be opened at the same time. You must load the MBASIC command file from CP/M in a different manner than was previously described. The specific instructions for running REPORT from CP/M will now be discussed. It should be pointed out that if you try to use the MENU program to run the REPORT program, an error message is produced and specific instructions are given. Thus, you cannot access the REPORT program from the MENU program. However, all is not lost. REPORT can be accessed from CP/M in the following manner:

1. Insert your Apple CP/M Accounting Package diskette into drive A and your current data file diskette into drive B. Power up the system as before.
2. When the Apple CP/M cursor appears enter from the keyboard the following:

```
MBASIC REPORT/F:4 <CR>
```

NOTE:

The MENU program is normally used to access the programs in the Accounting Package. However,the REPORT program is an exception. Apple CP/M will normally allow for three data files to be open at one time. Thus, when you type MBASIC<CR> from CP/M the system defaults to three data files being open at any given time within the program. The REPORT program requires four data files to be open and hence must be referenced as shown above. The **/F:** option allows you to specify or set the number of disk files that may be open at any one time during the execution of a BASIC program. If the **/F:** option is omitted, Apple CP/M automatically allows for three files to be open at any given time.

3. The Microsoft copyright will appear on the screen and then the system will load and begin the execution of the REPORT program.

The first screen is the "promo" screen and will appear as:

```
**** REPORT GENERATION PROGRAM ****
                by
      Steven M. Zimmerman, Ph.D.
            Leo M. Conrad
      Donald R. Smith, Ph.D., P.E.
*** (Press <CR> to Begin) ***
```

After pressing <CR>, the second screen will display the following as shown in Table 6.1.

```
          *** INSTRUCTIONS FOR REPORT GENERATION ***
This program generates an INCOME and BALANCE SHEET report using the
current TRANS file plus the following files:
     * old income/expense file
     * old balance sheet file
     * account number file
     * company name/address file
Data entry for this program follows:
```

**Table 6.1: Instruction Prompt for REPORT**

The user must answer five questions. To illustrate the application of the REPORT program we will continue with the same example problem that was constructed in the previous chapters. In general, the five questions pertain to the current date, the name of the current transaction file, the old income/expense balance file, the new income/expense file, and the disk drive containing the data files that are to be used.

The program first opens the files needed to input the company's name and address and then loads the old income/expense and balance sheet files. The current transaction file is loaded 1 transaction at a time to maximize the capacity of the program to handle transactions. There is no limit to the number of transactions the program can handle, but the amount of time needed to post transactions increases with the number of transactions.

The next section will review the sequence of entries required to generate these vital reports.

## 6.1 Running The Program

After the "promo" screen appears, a brief set of instructions and five specific questions are asked. The following discussion will utilize the example problem that has been presented in the previous chapters. The required user prompts are shown in boldface immediately after the prompt. For our example problem, we will respond with the following answers to the program prompts:

```
1. The Current Date (xx/xx/xx) is? 07/01/83 <CR>
2. Name of current TRANSACTION file with extender? MAY <CR>
3. Name of old BALANCE sheet file without extender? FIRST <CR>
4. Name of new BALANCE sheet file without extender? MAY <CR>
5. DISK (A,B,C,D) ? B <CR>
```

After the disk drive is specified the following will appear on the screen:

```
***** Opening the Files *****
```

The Balance sheet file extender = .BAL
The Income and expense extender = .INC

The transaction file carries the name "MAY" without an extender; therefore, we did not include an extender when asked for the name of our transaction file. The balance sheet file must carry the extender .BAL, and the income/expense statement must carry the extender .INC. The income/expense and

balance sheet files must have the same name except for the extenders. In the SETUP chapter we name the 2 files for the income and balance sheet "FIRST.INC" and "FIRST.BAL". In this program, the new income/expense file is named "MAY.INC" and the new balance sheet file is named "MAY.BAL".

After entering the responses to the five questions and pressing <CR>, you will note a brief delay of approximately twenty seconds or so. This is because the program is opening several files, reading the information, and conducting some calculations necessary to build the required reports. When the program is finished reading the files and the calculations, the following screen prompt appears:

```
**** FILE MANIPULATIONS COMPLETED ****
  Ready to PRINT to the line printer
ALIGN PAPER IN PRINTER AND TURN PRINTER ON
      (Press <CR> when ready) ?
```

This screen conveys the fact that all file reading and calculations are completed and you are ready to begin printing the reports generated by this program. By pressing <CR>, the program will begin the line printing process to your printer. Recall, our system was designed around the popular Epson MX 80 or MX 100 series of printers. The newer Epson FX series will also work. If you have some other printer, you may have to examine the BASIC listing and make some alterations of the print codes, or you may desire to customize the output formats according to your individual tastes.

When the line printer is finished printing the reports, the program will clear the screen and the following prompt will appear:

```
*** REPORT GENERATION COMPLETED ***
            Options Menu

Select one of the following:

(R)e-run this program again
(M)enu ...return to the main system menu
(T)erminate this run

Enter your choice (R/M/T) ?
```

If you enter an **R** <CR>, the program logic will branch to the beginning of the program and allow you to conduct another run. By entering an **M**, the REPORT program will clear itself out and return to the main system menu program. Entering a **T** will terminate the current run.

One important note before we examine the program—if you find an error after a run has been made, it is possible to rerun the program. A new income/expense file and balance sheet file will be created from the second run. If you use the same names as the first run, the first files will be OVERWRITTEN and therefore lost. Because of the overwrite feature of this program, you must be careful with the files of your accounting system.

The output produced from the REPORT program consists of two reports, an income (expense and income summary) and a balance sheet. Table 6.2

illustrates the income/expense report for the example problem, and Table 6.3 is the balance sheet for the example problem.

06/01/83
INTERNAL BYTES MANAGEMENT
123 SOLID STATE ROAD
ELECTRON, AL. 11111

INCOME & EXPENSE STATEMENT
Income

| Number | Account | Current Period Amount | Current Period Percent | Year to Date Amount | Year to Date Percent |
|---|---|---|---|---|---|
| REVENUE | | | | | |
| 6010 | RENT INC | 1,900.00 | 0.46 | 3,133.00 | 0.13 |
| 6020 | CONSULTI | 2,200.00 | 0.54 | 9,700.77 | 0.41 |
| 6030 | INTEREST | 0.00 | 0.00 | 788.45 | 0.03 |
| 6040 | SOFTWARE | 0.00 | 0.00 | 756.33 | 0.03 |
| 6050 | CAPITAL | 0.00 | 0.00 | 8,976.33 | 0.38 |
| 6900 | MISC SAL | 0.00 | 0.00 | 456.22 | 0.02 |
| | | $4,100.00 | | $23,811.10 | |

| Number | Account | Current Period Amount | Current Period Percent | Year to Date Amount | Year to Date Percent |
|---|---|---|---|---|---|
| EXPENSES | | | | | |
| 9010 | MAINTENA | 150.00 | 0.13 | 3,555.55 | 0.12 |
| 9020 | SUPPLIES | 0.00 | 0.00 | 1,203.44 | 0.04 |
| 9030 | DEPRECIA | 755.62 | 0.65 | 12,211.84 | 0.42 |
| 9400 | INTEREST | 258.00 | 0.22 | 7,603.44 | 0.26 |
| 9500 | OPERATIO | 0.00 | 0.00 | 1,645.00 | 0.06 |
| 9510 | UTILITIE | 0.00 | 0.00 | 218.66 | 0.01 |
| 9520 | INSURANC | 0.00 | 0.00 | 766.88 | 0.03 |
| 9530 | REAL EST | 0.00 | 0.00 | 566.77 | 0.02 |
| 9540 | BUSINESS | 0.00 | 0.00 | 521.00 | 0.02 |
| 9550 | OTHER TA | 0.00 | 0.00 | 122.00 | 0.00 |
| 9560 | PROF.SE | 0.00 | 0.00 | 97.00 | 0.00 |
| 9600 | TRAVEL | 0.00 | 0.00 | 123.00 | 0.00 |
| 9650 | MEDICAL | 0.00 | 0.00 | 489.00 | 0.02 |
| | | $1,163.62 | | $29,123.58 | |
| PROFIT | | | $2,936.38 | | -$5,312.48 |

TABLE 6.2: Income/Expense Statement for the Example Problem

INTERNAL BYTES MANAGEMENT
123 SOLID STATE ROAD
ELECTRON, AL. 11111

BALANCE SHEET

| Number | Account | Debit | Credit |
|---|---|---|---|
| 1 | CHECKING ACCT | 4,525.00 | 0.00 |
| 2 | PETTY CASH | 120.00 | 0.00 |
| 100 | ESCROW | 2,209.50 | 0.00 |
| 500 | ACCTS REC. | 15,000.00 | 0.00 |
| 1000 | FIXED ASSETS | 0.00 | 0.00 |
| 1001 | COMPUTER EQUIP | 27,928.62 | 0.00 |
| 1002 | RES FOR DEPR | 0.00 | 12,939.09 |
| 1030 | HOUSES | 180,656.48 | 0.00 |
| 1031 | RES FOR DEPR | 0.00 | 22,492.49 |
| 1400 | LOANS STOCKHOLDERS | 20,937.32 | 0.00 |
| 1500 | MORTGAGES | 52,627.30 | 0.00 |
| 1900 | LAND INVESTMENT | 59,936.83 | 0.00 |
| 1901 | ORGANIZATION INV. | 21.10 | 0.00 |
| | LIABILITIES | | |
| 2001 | ACCTS. PAY | 0.00 | 0.00 |
| 3999 | TAXES PAY | 0.00 | 425.60 |
| 4000 | SHORT TERM NO | 0.00 | 0.00 |
| 4010 | DEPOSITS | 0.00 | 736.50 |
| 4020 | SECURITY DEPOSITS | 0.00 | 1,042.36 |
| 4030 | SHORT TERM BUS. LOANS | 0.00 | 15,000.00 |
| 5000 | MORTGAGES | 0.00 | 281,975.59 |
| | STOCKHOLDERS EQUITY | | |
| 5998 | CAPITAL STOCK | 0.00 | 18,200.00 |
| 5999 | SURPLUS | 0.00 | 11,150.52 |
| | | 363,962.15 | 363,962.15 |

```
Computer files used are
     Transaction file: B:MAY
     Old balance sheet: B:FIRST BAL
     Old income/expense: B: FIRST.INC
     New balance sheet: B:MAY.BAL
     New income/expense: B:MAY.INC
```

**Table 6.3: Balance Sheet for the Example Problem**

## 6.2 Examining The Program

The program's identification is found in line 10. Line 20 clears the screen, provides space for string variables, and transfers to the program "promo" screen. Line 30 branches to a subroutine that prints a brief instruction paragraph on the screen. Line 50 prompts for the current date.

The remaining screen prompts are found in lines 90 through 120. Lines 130 through 170 combine the names of file information and the disk specification information into file names. Lines 180 and 190 print the informational statements about how the extenders are used.

The task of opening disk files to obtain the information needed to produce the reports required in the program begin in line 200. Specifically, the old balance sheet file is opened as file 1 in line 200, and the old income/expense file is opened as file 2.

In line 230 the transaction file is opened, and in line 240 the entire balance sheet file is read, while in line 250 the income/expense file is read. Line 260 reads a transaction, and the transaction is posted in lines 270 through 310. There is an error trapping routine in line 320 to check on possible errors in posting to nonexistent account numbers.

Lines 360 through 380 close the sum of the debits to the sum of the credits in each account in both the balance sheet and the income/expense statement. In line 400, the file that contains the information about which accounts are opened, and the information contained in the opened accounts is read in line 410.

Lines 440 through 490 move the periods transaction from the periods storage location B#(i,j) to A#(i,j) for inclusion in the new income/expense and balance sheet files. To clarify, the "#" symbol after a variable name means that the variable is a double precision variable. The next task the program performs is to start printing the reports needed. You now can see why there can be a long delay before the program starts the printing cycle. In line 500 a GOSUB to line 1710 prints a message on the screen telling the user that the file manipulations are completed and the printer is to be made ready for printing the reports.

Line 510 prints the date of the report. This is the date you entered when you started the system. (Make sure you use the date you want to have printed on the reports.) Line 520 prints a blank line, and line 530 starts the input of the name of your company from the file where it was recorded in the startup program. Line 550 is the actual input statement, and lines 560 through 580 are a subroutine to do the printing. We use a subroutine to print the name on both the income/expense statement and the balance sheet statement, using the same lines of code to perform both printing tasks.

Lines 590 through 600 print the title for the income/expense statement. Line 610 defines the name for the title accounts file and opens the file. The actual information is input in line 620. Notice that we are able to reuse the file numbers because we always close a file when we have completed the input or output task.

Line 630 starts the process of printing the income portion of the income/expense statement. In line 670 is an IF statement that picks out the title accounts to be used for that purpose. Be very careful you never post to a title account. It could result in an error difficult to find and correct.

Line 680 sets the sum of the income for both the period and year-to-date report equal to zero. Lines 690 and 700 actually add the amounts. Lines 720 through 830 are designed to produce the income part of the report.

Starting in line 840, the expense part of the report is produced. The coding is similar to the income part, and the task is completed in line 950. Summarizing information and posting of the profit for the period are performed in lines 1000 through 1070. The LPRINT instructions in line 1080 mark the separation of the income/expense statement from the balance sheet report.

The date on the balance sheet report is printed in line 1090, and the GOSUB instruction in that line prints the name of your company. Lines 1100 through 1130 print the titles needed by this report. Lines 1140 through 1230 are all the code needed to complete the balance sheet. Note: The title accounts are selected in line 1160 in a manner similar to that used in the income/expense report.

Lines 1240 through 1290 perform the task of printing the names of the computer files used to produce the report for the record. This information is very useful in keeping track of which file you use for what purpose. Line 1310 sends a forms-feed (CHR$(12)) to the Epson printer. Line 1320 through 1400 perform file handling chores needed by the program.

## 6.3 Summary

The task of generating the income/expense and balance sheet report is automatic. Once you have set up your business in the SETUP program and keyed in the transactions in the TRANSaction program, the REPORT program used the generated files to produce the required reports. You must be careful in selecting the correct transaction file, old income/expense and balance sheet file, and the name and the new income/expense and balance sheet files to make their use for the next accounting period an easy task.

At year-end, make sure you zero-out the income/expense statement information, so you may start with zero values in your earnings and expense-to-date file. The chart of accounts may be added to or deleted at any time, so if your business changes, the accounting system will be able to change with it. or deleted at any time, so if your business changes, the accounting system will be able to change with it.

# 7. Invoice Program

The task of producing an invoice represents one of the most important day-to-day business activities. Unless one's business is conducted entirely via cash transactions, it is necessary to produce a bill before getting paid. This program is designed to maintain a file of customers from which invoices may be produced, or to produce bills directly from the keyboard. If the customer file is used, the program will post to an accounts receivable transaction file using one overall accounts receivable number or individual account numbers.

The program operates from a series of menus from which the user may select alternate routines available in the program. The program is user friendly in the sense that although whole words are asked for in the menus, the program usually reacts to the first letter of the word only in the same manner as our other programs. For example, if one of the selections is START you may input START or Start or SaToNaTack. In all cases an S is in the first column. The exceptions to this rule are designed to occur where it is necessary for the user to be very careful to avoid costly errors. The program will accept both lower- and uppercase letters. For example, sTART, start, or sAtOnAtACK will control the actions of the pertinent routines.

This chapter is designed to give you two different levels of understanding of the program. The first level is as a user only. For an individual to use our programs, only a minimum knowledge of the internal workings of the program is needed. If you have an interest in using the program only, concentrate on the running the program section (7.2) and skim the details of the program. If you want to customize the program, you will need to understand all the sections.

The invoices produced by the program are all designed to be used with a window envelope to minimize the amount of typing and effort needed to complete the task.

## 7.1 Details Of The File System Used

The main purpose of the INVOICE program is to allow you to generate bills or invoices to be mailed to your customers who purchase goods or services from you. Therefore, one of the main functions of INVOICE is to initialize and allow you to maintain a file of customers. The program automatically names this file COMPANY.DAT. Three other files are required called INVNUM.DAT, NAMEX.DAT and trans.DAT. Before a bill can be sent to a customer, the party must be in the customer file. Also, the posting of the billing to the various accounts is handled by INVOICE so that the accounting system can function properly. As we progress through this chapter, we will describe how this posting is accomplished.

The customer file system is a random access file named COMPANY.DAT and includes the following data, allowing for the word size indicated:

```
String variables
*Customer's Name              12 characters
*Customer's Company           25 characters
*Number and Street            25 characters
*City                         15 characters
*State                         3 characters
*Zipcode                       5 characters
*First name                    8 characters
*Title                        10 characters
Numbers
*Current balance               8 characters Double-precision number
```

The customer's name, customer's company, number and street, city, state, and zipcode are used in writing the invoice. If you want to customize the data file, this may be accomplished with a minimum of effort. One of the stated objectives of this book is to provide you, the user, with a program that you may use as is or that you may customize for your own purposes. If you do customize the program, you must start with an understanding of the current file structure. To mark the end of this file, we use the procedure of inputting the word **END** for the string variables and the value **9999999999** for the number variables. This file is used and maintained in this program and the mailing list program. These are the only places where the file is used.

In addition to the above customer file, the program uses 3 other files. These files are:

```
INVNUM.DAT A sequential file used to save the invoice number and
  accounts payable number.
NAMEX.DAT A sequential file used to save the name and address of
  your company.
TRANS.DAT A sequential file to be named by the user to save the
  transaction data.
```

There are only three values in INVNUM.DAT. The first value is the invoice number, the second value is the number of the accounts payable account, and the third number is the sales tax account number. You must initialize the values stored in this file when you first start using the program.

The file NAMEX.DAT contains the following information:

```
Company's name
Company's address
Company's city
Company's state
Company's zipcode
Company's phone number
```

The above three files should be placed on disk B.

The transaction file also should be placed on drive B. It may carry any name you wish to give it. We suggest you name it for the time period it is associated with, as detailed in earlier chapters. The information in the file is as follows:

```
Transaction number as an integer number.
DATE$ as a string variable (a word).
```

```
Description$ as a string variable.
Account debited as an integer number.
Account credited as an integer number.
Amount debited as a double-precision number.
Amount credited as a double-precision number.
Clear$ as a string variable to indicate if the transaction has
  cleared the bank.
```

To mark the end of this file, we input the word **END** for all string variables and the value **0** for all numerical variables.

## 7.2 Running The Program

To execute the INVOICE program, insert the program diskette in drive A and bring up CP/M. From CP/M type **MBASIC/F:4 <CR>**. Apple CP/M will bring up the standard Microsoft logo and then begin the execution of the program. The first screen is the "promo" screen:

```
**** INVOICE GENERATION PROGRAM ****
                by
      Steven M. Zimmerman, Ph.D.
            Leo M. Conrad
      Donald R. Smith Ph.D., P.E.
       ( Press <CR> to Begin )?
```

The program starts by displaying a short set of user instructions. The instructions state:

```
THIS PROGRAM IS DESIGNED TO MAINTAIN A FILE OF CUSTOMERS FROM WHICH
  INVOICES MAY BE PRODUCED, OR TO PRODUCE INVOICES FROM THE KEY-
  BOARD. THE PROGRAM OPERATES FROM A SERIES OF MENUS. THE FILING
  SYSTEM USED BY THE PROGRAM IS A RANDOM ACCESS SYSTEM. IT IS
  SUGGESTED THAT YOUR CUSTOMER FILES BE MAINTAINED ON A SEPARATE
  DISK FROM THE ACCOUNTING PACKAGE. HOWEVER, PLACE A COPY OF 'NA-
  MEX.DAT' FROM THE SETUP PROGRAM ON THIS DISK. THE PIP UTILITY CAN
  BE USED TO COPY THIS FILE FROM THE ACCT. DATA DISK TO THE IN-
  VOICE DATA DISK. IF YOU HAVE A 2-DRIVE SYSTEM, USE DRIVE B FOR
  INVOICE DATA.
```

To illustrate the application of INVOICE, we will assume that a new customer, Mr. Bob Jones, has placed an order with your small company for a microcomputer software package that costs $145.00 per copy. Assume that he has requested two copies of the system. Step through the program to see how an invoice can be generated on the line printer. We assume that you have a line printer and a printer interface card in slot 1 of the Apple.

You will generate data as you proceed through. The first prompt is:

```
1. BILLING DATE (XX/XX/XX) IS: 07/15/83
2. Disk Drive for Data (A,B,C,D) ? B
3. Transaction file (Y/N)? Y
```

You have the option of creating a transaction file using individual customer's numbers, or a file with one overall account receivable, or not creating a file at all. If you elect to use a file, the next question is:

```
4. Post to (I)ndividual number or (A)ccounts rec. no.? A
```

It was decided to post to the overall accounts receivable number in this example because you do not as yet have a customer file established on our data diskette. Later on in this chapter, you will be shown how to establish a customer file.

The next question asks the name of the transaction file. These next two questions are shown with our answers for the sample run:

```
Name of file? INV.MAY
```

The choice of the name "INV.MAY" is arbitrary and is selected by the user. However, we suggest that you use the file extender "XXX" where XXX is a particular month abbreviation. In this manner, you can recognize your invoice files and the month that they apply.

As soon as you press <CR>, you will note that the drive light on drive B will come on, indicating that the file INV.MAY has been opened and data recorded on the diskette. When the disk light goes out, the main menu for INVOICE will appear on the screen. This menu directs you to the type of printer output desired or the task of working with the data base of customers for your system. The menu reads as follows:

```
Select output types:
     (F)ILE of customers
     (M)X-80 Printer
     (P)re-printed forms
     (O)THER printers
     (E)nd
```

Assume you want to produce an invoice from the keyboard using the Epson MX-80 printer. Next to the question mark above, type MX- 80 or simply **M** and press <CR>. The form used for the printed form option may not fit your needs exactly, so you will need to customize this part of the program. The (O)THER option will produce an invoice using almost any printer, as long as it can handle 80 columns of output.

The next question relates to the starting number to be used on your invoice. You must initialize this number. If you do not, the program will not work correctly. When you start the system, the file that carries the information relative to invoice number does not exist on your disks.

The question reads:

```
New starting number (Y/N)? Y
```

Assuming you answered yes to the above, the next prompt sequence will be in a tabular form as follows:

| STARTING NO. | ACCTS. REC. NO. | SALES TAX NO. |
|---|---|---|
| 215 <CR> | 500 <CR> | 6900 <CR> |

Some business persons may want to continue the numbering series currently in use in their business, while others just may not want to start their

invoicing number at one. We chose to start with number 215. Type 215,500,6900 for the invoice number, the accounts receivable number, and the sales tax number. Recall from the SETUP program, the accounts receivable number was account no. 500. Since you are posting to the general account, accounts receivable, you must specify the correct number. If you forget this number, refer to the printout of your income and expense report that was produced by the REPORT program.

If you get an error message at this point, it is likely that a file cannot be found. This means you have not used the SETUP program to initialize the name of your company. Go to the SETUP program, initialize the name of your company, and start over.

The next question relates to the method of input for the customer's name and other information. You may select to input the customer's information directly from the keyboard or from a data file. For this run, an example will illustrate how the direct keyboard method works. Since you do not as yet have a customer file created on our invoice diskette, you cannot use the (D) option until such a file exists. You will be shown how to create this file later in this chapter. The question and answer follows:

```
You must use disk file input to post to individual accounts in the
transaction file.

Select input method -- keys or disk (K/D)? K <CR>
```

The reason chosen to use the overall accounts receivable number was to use keyboard input. The customer must have a number to post data to a specific account, and the system does not assign the person a number until they are entered into the customer file. The customer's data questions and answers follow:

```
Input Customer's Name                    ? Jones <CR>
Input Company's Name                     ? Jones Smith and Jones <CR>
Input Company's Address                  ? 123 J Street <CR>
Input City                               ? Jonesville <CR>
Input State                              ? AL <CR>
Input Zipcode                            ? 11111 <CR>
Input First Name                         ? Bob <CR>
Input Title                              ? President <CR>
Input Current Balance                    ? 0 <CR>
```

If you had decided to answer the question relative to source of input information with **D** (disk), we would have seen the following:

```
The menu for file input is as follows:

File Input Menu
-1   File number
 0   Customer's name
 1   Company's name
 2   Number and street
 3   City
 4   State
 5   Zipcode
 6   First name
 7   Title? -1
```

The system is designed to search for a particular file number to a specific title of an individual. In this run, the assumption is that the bill will be sent to customer number 1, and input as -1, so that you could search using this method. The next question and answer consist of:

```
File - Customer Number? 1 <CR>
```

In response to your answer, the computer selects customer number one and prints the information below:

```
Number              Title                    Input
0                   Customer's name          Jones
1                   Company's name           Jones Smith and Jones
2                   Number and street        123 J Street
3                   City                     Jonesville
4                   State                    AL
5                   Zipcode                  11111
6                   First name               Bob
7                   Title                    President
8                   Current balance          0
99 to continue, 98 to search more, or number to change? 99
```

(If you had continued at the keyboard, you would be at this exact same point in the program.) The options following the information refer to the task of searching the data files to find a particular customer. If you select option 98, you will be able to page through the entire data file. If you select 98 while entering data from the printer, you will get an error message.

The controls allow you to change any of the above listed information or to continue to the next part of the program by typing **99** and depressing <CR>. Regardless of which option you select, you will end up with the next prompt:

```
*** Input INVOICE data Section: To Stop, type 'END'
Sales tax rate as percent (o if none)? 6 <CR>
```

If you are charging your customer sales tax, the tax rate is typed here. If the rate is 6 percent, then you would type **6** as the answer to the above question. The next question is:

```
Purchase order number?
```

Many items are sold by purchase order number. Enter the purchase order number here. If no purchase order number is used, press <CR>. For the example, assume the PO number is **475**. After typing **475** followed by <CR>, the program sets up a tabular input format for entering the customer's order. The screen format is as follows:

```
ORDER DATE    ORDER DESCRIPTION    To end, type END <CR> END <CR>
?
```

Be careful to enter the date in the format 00/00/00 or in any other manner that does not require the use of commas. When you use a comma, the computer will assume that you have completed the input of the date and that the

additional information is the line item. For your sample run, answer the above question as follows:

```
   ORDER DATE          ORDER DESCRIPTION      To end, type END <CR>
                                              END <CR>
?12/31/82 <CR>      ?Q. C. Package <CR>
```

The next tabular prompt sequence is:

```
PRICE/UNIT     QUANTITY ORDERED
?
```

Answer these questions with **145.00** <CR> **2** <CR> to indicate the price of the item was $145.00 each, while the quantity delivered was two items. At this point, the program will return with the original tabular prompt shown below:

```
ORDER DATE     ORDER DESCRIPTION    To end, type END <CR> END <CR>
? END <CR>     END <CR>
```

Note the use of "END" typed under each column provides for the termination of the sequence. This is one of the exceptions to answering a question with the first letter of a choice only. You must type either end or END to get the computer to terminate the above cycle. This is necessary just in case part of the word end was being used for some purpose.

The computer terminated the above input series and printed the following on the screen for potential correction purposes:

```
NO.   Date   Item Description   Price/Unit   Quantity   Extended Total
0     12/82  QC Package            145.00          2        $290.00
                Sales Tax                                     $17.40
                                                             $307.40
Input 99 if complete or number for change? 99
```

Assuming you answered 99 to the above question, the next question is:

```
Default terms (Y/N)?
```

Built into our program is a set of default selling terms. These terms should be adjusted for your business. This option allows you to input special terms for individual cases. If you answer **Y**, the system will print the default terms **"One percent, Ten days, net thirty, 1.5% per month late charge on remaining balance at the end of month."** If you answer **N**, the program will ask you to type in your own terms as an alpha-numeric string.

The next prompt asks you to make sure your printer is properly set up and turned on. Make sure paper is loaded correctly and the printer is set to the top-of-form. For the Epson printer, you must make sure the printer is off before you adjust the paper to the top of the form. When you have finished this adjustment, turn the printer on. The printer's internal memory will "remember" this setting. This prompt reads:

```
<CR> when printer is ready?
```

It is common to get so involved in a program that you forget to set up the printer. This prompt is designed to remind you to do this task. When you

press <CR> as the answer to the above question, the computer will produce the invoice as specified.

Table 7-1 shows the invoice produced by this program.

```
Internal Bytes Management, Inc.
Accounts Receivable
123 Solid State Road
Electron, Ala lllll

lll-lll-llll

**********************************    Invoice 215 Date: 07/15/83
* Jones                          *    Purchase Order #: xxxxxx
* Jones Smith and Jones          *
* 123 J. Street                  *
* Jonesville, AL lllll           *
**********************************

*********************************************************************
*    Date     Description     Price     Quantity    Sub-total      *
*********************************************************************
*    12/31    QC Package    $145.00        2          $290.00      *
*               Sales Tax                               $17.40      *
*********************************************************************
                              Total                    $307.40
                                                  ********************

One percent Ten days, net thirty, 1.5% per month late charge on
remaining balance at end of month
```

**TABLE 7-1: Specimen Copy of the INVOICE Program Printed Invoice**

When the invoice is finished printing on the line printer, one additional prompt appears on the screen.

```
INPUT sales income account number?
```

This prompt must be answered very carefully. As you recall, we selected the "A" option, i.e., we specified that we were going to post to an account. The program assumes that you are posting to an existing account in the chart of accounts that has been established via the SETUP program. Examination of the income/expense report reveals that we have an account called "Software Sales," account number 6040. Thus, we must specify this account number in order for the proper posting to take place. Therefore, respond with **6040** <CR> to this prompt.

After pressing <CR>, the program will return to the main driver menu:

```
***** MAIN MENU - INVOICE PROGRAM *****
Select output type:
     (F)ILE of customers
     (M)X-80 Printer
     (P)re-printed forms
     (O)ther printers
     (E)ND
Enter your choice ( F / M / P / O / E / )?
```

This time, select the FILE option. After pressing <CR>, you will see the following:

```
File Maintenance Menu

(A)DD Add new customers
(B)ROWSE the file
(D)elete file
(E)nd
(S)TART new file
(U)PDATE existing file
```

You must use the START option at the beginning of a new file. This option will place end of file markers (the word END) in all string storage locations and the value 9999999999 in the number storage locations. This information tells the program when the end of the file is encountered. (Our example of running the program will not work until you have input the first customer in your customer file.)

When a customer is deleted from the customer file, the word OUT and the value 9999999999 are written throughout the file. A new customer will be allowed to reuse an old customer number. This uses the disk storage space most efficiently.

The UPDATE option allows you to change any and all information in a customer's file. The procedure used is the same change option used when the invoice was being printed.

The ADD option will add new customers into the file at the end of the file or in place of deleted customers. The program is designed to use the lower numbers first.

Starting from the file maintenance menu:

```
File Maintenance Menu
     (A)DD Add new customers
     (B)ROWSE the file
     (D)elete file
     (E)nd
     (S)TART new file
     (U)PDATE existing file

Enter your choice ? S <CR>
```

In other words, we will select the "S" option for starting a new customer file. The program will generate the following warning prompt:

```
Are you sure you wish to continue?
```

To answer this prompt, you must type completely . . . YES or NO.

In this case you must type **YES** in all capital letters to continue. This is to make sure you do not destroy a file in error. As already noted, in most cases you are able to answer all questions with the first letter of the option in either upper- or lowercase. This is one of the exceptions to help protect the user from making a very serious error.

Once you have returned to the file maintenance menu, it is possible to start adding information to the file. You must answer 9 questions, from the

customer's name through to the current balance due. After you have input the data, you will have the opportunity to correct any errors you may have made.

For example, the correcting option looks like the following:

```
Number                Title                          Input
0                 Customer's name                Stanley Ande
1                 Company's name                 Anderson and Anderson
2                 Number and street              123 West Drive
3                 City                           Mobile
4                 State                          AL
5                 Zipcode                        11111
6                 First Name                     Stanley
7                 Title                          Manager
8                 Current balance                1093.98
99 to continue, 98 to search more, or number to change? 99
```

This is the same information as was input into the program when you produced an invoice from the customer data file. You will note that the customer's name has not been used very well. This was to show you what may happen if you do not carefully plan how you are going to use a file. We suggest that in the customer's name position you save only the last name of the customer. You will have to work this out for your particular needs.

After typing **99** and pressing <CR>, you will see the file maintenance menu. This time, select the UPDATE option. You will see the following:

```
Search Menu
-1    By file number
 0    Customer's name
 1    Company's name
 2    Number and street
 3    City
 4    State
 5    Zipcode
 6    First name
 7    Title
 8    Current balance?
```

Notice that item number 8, current balance, may be used to search for a particular file in addition to the 9 items used when you were producing an invoice from the data file. In the case of option -1, you must know the file number you are seeking. In 0 through 7, you may search with a part of a name. In 8, you may search for an exact amount or for files where more or less than some other value is owed. This customer's file is the same one used in the mailing label/envelope program. It may be maintained here or in the mailing program.

The (B)ROWSE option is a very useful feature of the INVOICE program in that it allows you to examine all of the customer files on the screen or provide a file-by-file listing on the line printer. To use this feature, select the

B option from the file maintenance menu by typing **B** <CR>. The (B)rowse option menu will appear as:

```
                    BROWSE CUSTOMER FILE SECTION
                   **** Browse File Options ****
(1) - SCREEN DISPLAY ONLY
(2) - LINE PRINTER SUMMARY ONLY
(3) - RETURN TO MAIN MENU
                    Enter your choice (1/2/3) ?
```

Selecting option 1, "SCREEN DISPLAY ONLY" will provide you a partial listing of the record of each customer file. By a partial listing we mean that only the file number, customer's last name, company's name, city, and state are shown across the screen. The remaining elements of the customer's record are not shown due to space limitations on the screen. Assuming you have set up a customer file, the screen display option, option **1**, would yield the following screen format:

| File No | Cust. L. Name | Company's Name | City | State |
|---|---|---|---|---|
| 1 | Adams | Adams & Assoc | Brookville | AL |
| 2 | Baker | Sysco, Inc | Evansville | MO |
| . | . | . | . | . |
| . | . | . | . | . |
| . | . | . | . | . |
| 56 | END | END | END | END |

The above display contains purely hypothetical data to illustrate the screen layout. Note that the last record in the customer file is marked in the file and displayed on the screen with the word "END". This shows you the last record in the file. It is important to point out that the customer file is not sorted in any particular manner and equally necessary that you know or can obtain the file number of a particular customer for deletion or update purposes. The browsing option allows you to see the customer sequence on the screen.

Option number 2, the line printer option, allows you to list on the printer the entire contents of your customer file. The program logic is set up to list four customers per page. You will find this option useful in that it allows you to maintain a hard copy of your customer file.

Two remaining options contained in the main menu have yet to be examined. They are the (P)re-printed forms option and the (O)ther printers option. The "P" option allows you to customize the program to print on pre-printed forms that are fed into your line printer. To do this, you will have to go into the BASIC program and re-format the various print statements in the pre- printed forms portion of the code. The next section will examine the code and show you where the changes will have to be made. However, to accomplish this customizing, knowledge of the MBASIC language is required.

The (O)ther printers option is similar to the MX printer option except for the fact that certain special printer codes have been omitted in order for almost any printer to function with the program. Again, if you have knowledge of

MBASIC and of the special features of your particular printer, you can customize the output as you see fit.

The next section of this chapter will review the details of the program, section by section. If you plan to change the code to suit individual requirements, you will have to understand how the program functions.

## 7.3 Examining The Program

The program is divided into 6 main parts. Part 1 consists of the main menu and procedures for initializing the invoice number counter, inputting the name of the billing company, inputting the company's information that is being billed from the keyboard, and inputting the invoice information relative to date, item, price per unit, and quantity. This part of the program starts in line 10 and ends in line 610.

Part 2 of the program is designed to input data from the customer's data file. This is an optional method of data input to the invoice. This part of the program starts in line 620 and ends in line 760.

The third part of the program consists of the housekeeping functions of entering data to the customer's data file, updating the file, deleting customers from the file, and similar activities. This activity starts in line 770 and ends in line 1280.

The fourth part of the program is designed to print an invoice on the MX-80 printer. This routine takes advantages of many of the graphics abilities of this printer. It starts in line 1290 and ends in line 1620.

The fifth part of the program is for printing an invoice on a regular printer without any special graphics abilities. If you are using an MX-80, a daisy wheel, or any other type of printer, this routine will work. The routine starts in line 1630 and ends in line 1910.

The last part of the program starts in line 1920 and is designed to produce an invoice using a commercial form. You may want to customize this program to fit the form selected by your company.

Starting in line 2150 is a series of subroutines used by the other parts of the program. The first subroutine in this set posts to accounts receivable when needed. The next subroutine handles the rounding problems before storing the information in data files.

The subroutine beginning in line 2500 and ending in line 2840 is the browse option portion of the program. This subroutine is designed to provide an on-screen browse option for all customer data files contained on drive B. Also, the line printer option for printing customer files is in this subroutine.

Lines 3000 through 3180 (called from line 20) are the introductory screen "promo" for the program. Lines 4000 through 4010 represent a short subroutine used by various parts of the program to provide an on-screen format for displaying data.

## 7.4 Customizing The File Structure

The data file structure must be established before it is possible to operate the invoice program. Lines 790 through 800 are designed to open the file for both input and output of data. The variable used to transfer information into or out of the file is called CCUST$(i). Line 690 defines the sizes of each of the variables being used. There are eight string variables and one number variable. You may change the size of any one of the fields in this field statement. You must do so before you start using the file. If you want to add data to this file, it has been made easy for you to do so, as long as you keep all the string variables in the front of the file and all the number variables toward the rear of the file. In line 60, NF is the number of string variables and NNUM is the number of number variables. You must change these one numbers and line 790 to change the number of fields in each data record. Be sure to read carefully the format associated with the field statement for random files before making your changes. In addition to the changes already noted, if you add fields, you will have to add to the field identification information in lines 70, 80, and 90.

The variables LEO$(i) and LEO(i) are used throughout the program to transfer information to and from the input/output buffer (stored under the name CCUST$(i)) or to the buffer. If you decide to increase the number of items in the buffer beyond 9, it will be necessary to dimension LEO$(i), LEO(i), and CCUST$(i) in a DIMension statement. There is a DIMension statement in line 120, and these three variables could be added to the list if desired.

The random file COMPANY.DAT is divided into two parts (the first part for string variables, and the second for double-precision numbers) because of the manner in which information is stored in this type of file. All information must be string variables. To convert a double-precision value to a string variable and back again, use the CVD and MKD$ functions. You must not use these functions when dealing with the string variables that actually are string variables. The intent of this book is not to be a tutorial in BASIC, so we will not go into detail on how to use these abilities. Study these ideas carefully before you try to customize the file system.

## 7.5 Eliminating The Reuse Of Customer Numbers

Notice that in line 810, the program searches for the words OUT or END to find an empty slot in which to place a new customer. The word OUT is used to delete a file and tells you that the space and customer number are available for use once again. This means you will reuse customer numbers as customers are replaced. You may not wish to follow this practice. If you do not, remove the OR LEFT$(K$,1) = "OUT" from line 810. This means you will someday fill all available storage space with old customers, but you will not use the same number for two different customers at different times.

In lines 120 and 130 in the first part of the program, the program uses the variables LEO$(i) and LEO(i). If you change the use or number of LEO$(i)s or LEO(i)s, it will be necessary to make adjustments in these lines. A change in line 20 of the values of NF and NNUM is all that is needed.

## 7.6 Customizing Terms Of Sale

To customize the terms of sale to your particular needs, it is necessary only to change the words between the quotes in line 600. Take care to print some sample runs to make sure the message does not index in the middle of a word. You may have to add some extra spaces in the message to make it come out correctly using the compressed print of the MX-80 and the regular size print of any other printer.

## 7.7 Deleting The "Get Printer Ready" Message

The get printer ready message may not be needed in your operation if you leave the printer on at all times. To remove this message, change line 610 to:

```
610 GOTO 1300
```

This will speed the operation of the program a little.

## 7.8 Making Adjustments For Your Customized Form

In line 1930, there is a variable called LEFT. Change the value of this variable to move the output from left to right on your printer. In line 2110, change the number of LPRINTs to adjust the spacing for the form you have selected. You may wish to replace all of these LPRINTs with a LPRINT CHR$(12) to advance to the top of the next form.

## 7.9 Summary

This invoicing program is designed to produce three different invoices on the MX-80, or on any printer including, but not limited to, a daisy wheel printer, and on special forms. Because of the large variety of forms available, fitting the output to a specific form is left to the user. You may customize the form part of the program to fit the form that best suits your needs.

The form, as designed, uses envelopes with windows to minimize the amount of work associated with producing the invoice. The invoice was customized for a small business that sells goods or services in various quantities and must charge sales tax on its services.

In addition to customizing the form, we have reviewed many of the internal workings of the program and have added a large number of REMarks statements to help you customize the program for a particular need. The most difficult part of the program to customize is the file system, and we suggest that you take great care if you decide to perform this task. We have used the same file system developed here as the basis of our mailing list program and as part of the general ledger program, so take care to note the consequences if you decide to start making changes here. All the programs must be changed if they are to work together as a package.

# 8. Mailing Label Program

This program is designed to work with the database built in either the check-writing program or the invoice program for preparing mailing labels, customer/supplier lists, or envelopes. In operation, the file maintenance part of the program is similar to the other programs, but additional capabilities, such as sorting, have been added. Because of the similarity between this program and others, only a minimum amount of instruction will be included in this chapter. You should refer to the chapters on invoicing and checkwriting for additional details.

It may be necessary to make some program changes in this routine to match specific sources of commercial labels and envelopes. The program is formatted to handle a general-purpose gummed mailing label commonly found in office supply shops.

## 8.1 Running The Program

The program begins with the following heading and credits:

```
**** MAILLIST GENERATION PROGRAM ****
                by
        Steven M. Zimmerman, Ph.D.
              Leo M. Conrad
        Donald R. Smith, Ph.D., P.E.

        ( Press <CR> to Begin ) ?
```

After pressing <CR>, the screen will display a brief explanation of the program with some insructions regarding the data diskette to be used by the program. This screen states:

```
            ***** MAILLIST GENERATION PROGRAM *****
-PURPOSE OF THIS PROGRAM:
```

This program will assist in the generation of mailing labels or for printing names and addresses on envelopes. This program is designed to work with the database used by the check writing program and the invoice program. The program requires that a file of either customers or suppliers be maintained on a diskette in drive B. Place your file diskette in drive B and press <CR> to begin.

```
PRESS <CR> TO BEGIN ?
```

After a short delay, the program will prompt with the following:

```
1. (C)ustomer file or (S)upplier's file ? C
```

To illustrate, we will specify the **C** option. The next prompt is:

```
2. Disk (A,B,C,D)? B
```

We assume the file is or will be placed on the diskette in drive B. Type **B** and you will see the output menu:

```
Output Menu
     (E)nd
     (F)iles
     (L)abels
     (M)ailing envelopes ? F <CR>
```

If **F** is typed, the file mintenance menu will follow:

```
File Maintenance Menu
     (A)dd new customer
     (E)nd cycle
     (D)elete file
     (L)ist
     (P)rint list
     (O)rder
     (S)tart new file
     (U)pdate an existing file
?
```

The file may be listed, printed, ordered (sorted), or have any of the other actions taken on it that are available in the invoice and checkwriting programs.

Type **S** <CR> to clear the file. If you have already created a customer file, **this option should not be selected!** Assuming you do not have such a file, type **S** <CR> and you will see:

```
ARE YOU SURE YOU WISH TO CONTINUE? Type completely ... YES or NO?
```

This is the same protection routine used in other programs to prevent you from destroying files in error. Type **YES** <CR> ONLY if you do not have a file started. You will now see the file maintenance menu:

```
File Maintenance Menu
     (A)dd new customer
     (E)nd cycle
     (D)elete file
     (L)ist
     (P)rint list
     (O)rder
     (S)tart new file
     (U)pdate an existing file
?
```

Select (A)dd by typing **A** <CR> to add a new customer to the file and you will see:

```
Input last name ?
```

This is the first of a series of questions designed to enter the data needed into the customer/supplier file. A complete set of questions with sample answers as:

```
Input last name                    ? Anderson <CR>
Input company's name               ? Smith and Johnson <CR>
Input number and street            ? 123 West Drive <CR>
```

```
Input city                          ? Mobile <CR>
Input state                         ? Alabama <CR>
Input zip code                      ? 11111 <CR>
Input first name                    ? Tom <CR>
Input title                         ? Quality Control Manager <CR>
Input current balance               ? 0 <CR>
```

The computer produces a listing for you to check and correct as needed. The screen format of this listing is:

```
Number              Title                       Input
0                   Last name                   Anderson
1                   Company's name              Smith and Johnson
2                   Number and street           123 West Drive
3                   City                        Mobile
4                   State                       Alabama
5                   Zip code                    11111
6                   First name                  Tom
7                   Title                       Quality Control Manager
8                   Current balance             0
99 continue, 98 search, or number to change?
```

If you type **99** <CR> you will be returned to the file maintenance menu. If you type 98 you might get an error. This option should not be used unless you start with the (U)pdate option in the file maintenance menu.

**SPECIAL NOTE** ********** If you happen to make an error and use the (S)tart option when you should not, the update option allows you to enter the data file and redefine any record. Take care to leave at least one trailing END record.

The (O)rder option allows you to sort the names and addresses in your list in any order, according to any field in the record. After selecting (O)rder from the menu, the Order menu appears. Following your selection of fields, there will be a delay while the sorting is performed. When the sort is complete, the program automatically returns to the program menu.

## 8.2 Examining The Program

The program is built on a series of routines to perform the tasks called for by the menus. The first part of the program, from line 10 to line 200, names the program, initializes the value of the string variables used for headings, and prints the heading, the credits, and the first question.

Lines 210 through 230 complete the input of initial information, and then send the computer to the subroutine in lines 580 through 610 to open the data file and to define the method of organizing the fields in the file.

Lines 240 and 250 are the output menu, while lines 260 through 300 route the program to various parts of the program, depending on the selection made in the menu.

Lines 310 through 410 are used for a subroutine that inputs the file information. At the end of the routine in line 380, you will find the question you see at the end of inputting a file.

The routine that searches for a given data file is located in lines 42 through 560. Since it is possible to search by many different items, this routine was designed to handle these different cases.

Line 640 begins the file maintenance menu. This menu is found in lines 650 through 700. Lines 710 through 790 are then used to route the program, depending on the menu selection.

Lines 800 through 870 are the add option, while lines 880 through 1070 contain the delete and update options. Lines 1080 through 1140 house the code used to initialize the data file.

Lines 1150 through 1290 are an additional part of the search routine. This code is needed to find files with specific dollar amounts in the account due or payable.

Envelopes are produced by lines 1300 through 1360. You may have to do some custom work here if your envelopes do not match the ones for which the program is formatted. The value of LEFT, in line 1320, is changed to move the address left and right, and the number of LPRINTs in line 1310 is changed to move the address up and down.

Labels are produced in lines 1370 through 1430. The number of LPRINTs in line 1390 adjusts the distance between addresses.

Lines 1440 through 1640 are for the printing option, while lines 1650 through 1850 are used to order the names as requested. Lines 1860 through 1900 are a subroutine used to print file information on the monitor.

The input of the lines to be listed is entered in line 1910, and the listing routine continues through line 1990. Lines 2000 through 2170 perform the task of printing a list of customers on your printer.

## 8.3 Summary

This program is a very useful utility for assisting in maintaining a simple database of customers and/or suppliers. The program can assist in mass mailings of your firm's advertising or for whatever purpose you deem necessary. The menus are easy to use and the program is modifiable to suit your individual needs.

# 9. Checkwriting Program

The checkwriting routine is written to produce checks from a data file of transactions and suppliers or to work directly from the keyboard with or without a file of suppliers. When set to work from the keyboard, the program can produce a transaction file used in the transaction program. The transaction routine is able to merge the transaction file produced in the checkwriting routine with other transaction files.

The checks produced by this program are designed for a computer system with a printer with tractor feed and at least 80 columns. The assumption is that you will be purchasing custom checks for your system. (It is not possible to produce a check on a blank piece of paper because of the need for magnetic markings on the check.) We will help you adjust your output to the particular check you select. The program is now set for the system.

## 9.1 Running The Program

To run CHECK, you must enter MBASIC from Apple CP/M with the following:

```
MBASIC CHECK/F:4 <CR>
```

The program starts with the following credits and question:

```
**** CHECK WRITING PROGRAM ****
                by
      Steven M. Zimmerman, Ph.D.
            Leo M. Conrad
      Donald R. Smith Ph.D., P.E.
        (Press <CR> to Begin)?
```

After pressing <CR>, the screen will clear and the following screen display will appear:

The checkwriting program is designed to produce checks from a data file of transactions and suppliers or, to work directly from the keyboard without a file of suppliers. The checks produced by the program require that custom checks be used with a tractor- feed printer. Your printer must have tractor-feed capability.

If you are going to use this feature, install the checks in your printer and align to top-of-form.

```
Press <CR> to Continue
```

After pressing <CR>, the screen will clear and the first data prompt will appear as:

```
1. Today's Date is ? 6/15/83 <CR>
```

The next prompt is:

```
2. DISK Drive for data files (A/B/C/D) ? B <CR>
```

This question relates to which disk drive will be used to save the files. The **B** answer to the question tells the computer to perform the save routines on drive B. After entering B <CR>, the drive light on drive B will come on for approximately 1 second and the Transaction File Menu will appear as:

```
                ***** TRANSACTION FILE MENU *****
(I)nput
(O)utput
(N)one ? O <CR>
```

You can choose either a transaction file for input or output, or not use any file. (It makes no sense to have both an input and output file with 2 different names that must be identical.) We have selected the option of an output file. The plan is to use the output file as input in some future run to illustrate how the file must be designed and what information goes where. The transaction file, as used in the program, is identical to one that could be produced by the transaction program.

The transaction file produced may be used in the transaction program for correction purposes, or it may be merged with other files to be used in the report program.

The next question relates to the name of the transaction file:

```
Name of file? CHECK.MAY
```

The names of transaction files should be tied into the period in which the transactions occurred. We used CHECK.MAY in this example. The reverse MAY.CHK would have worked just as well. The choice is left to you. Be careful not to use the same name for 2 different files. One will wipe the other one off the diskette.

The next question relates to the type of output desired.

```
SELECT OUTPUT TYPE:
    (F)ile update
    (C)heck writing
    (E)nd
    ? C <CR>
```

You can update the customer file to produce checks or end the program. In this and the INVOICE program, it is very important that you go through the ending option. The file closing task is performed only when this option is selected. Some files must remain open while the program is being used, although we can close some.

The program continues with the following question:

```
New starting number (Y/N)? Y <CR>
```

You must initialize the check starting number, the lowest number in the accounts payable file, the highest number in the file, and the cash account

number. Once initialized, you will be able simply to answer "N" to the above question. After responding with a "Y" to the above prompt, the following tabular prompt will appear on the screen:

| Start.CK.No. | Accts Pay (Low) | Accts Pay (High) | Cash Acct.No. |
|---|---|---|---|
| ?215 <CR> | 2001 <CR> | 3999 <CR> | 1 <CR> |

If you are posting transactions to suppliers, it is necessary to use individual accounts. If you are working from the keyboard, it is not necessary to post to specific accounts, but it is possible to post to some overall accounts payable account. To post to specific accounts, you must know the account number, which, in our system, starts with the lowest numbered accounts payable account and ends with the largest. Using the system as designed, it is possible to have 3999-2001 or 1998 accounts payable customers.

Checks must be written against some cash account. The program must know what the number is, so it may do the posting. We have used account number 1 for the cash account.

The program proceeds to the next question:

```
You must use disk file input to post transactions to individual
accounts.
Select input method-- (K)eys or (D)isk ? K <CR>
```

Use the keyboard as the input for the first run.

The next set of questions and answers is:

```
Input Supplier's Name                    ? Anderson <CR>
Input Company's Name                     ? Sara and Rower <CR>
Input Number and Street                  ? 123 Beller Blvd <CR>
Input City                               ? Retailville <CR>
Input State                              ? AL <CR>
Input Zipcode                            ? 11111 <CR>
Input First Name                         ? Sam <CR>
Input Title                              ? President <CR>
Input Balance Due                        ? 12.33 <CR>
```

```
Number                  Title                        Input
0                   Supplier's Name              Anderson
1                   Company's Name               Sara and Rower
2                   Number And Street            123 Beller Blvd
3                   City                         Retailville
4                   State                        AL
5                   Zipcode                      11111
6                   First Name                   Sam
7                   Title                        President
8                   Balance Due                  12.33
99 to continue, 98 to search more, or number to change? 99 <CR>
```

The screen listing of the input data is provided for you to check your work and determine whether any changes are necessary before proceeding. You may correct anything from supplier's name through balance due. Option 98 may be used to search data files, but you will make an error if you select it at this time.

Assuming you answered "99" to the above prompt, you will see:

```
Input invoice number ? 99999
Purchase order number? 123455
```

As designed, the program prints the invoice number as the identification number for the supplier when input is from the keyboard, and the check or transaction number when information is from a file. We will point out where action is taken and show how you may change the number printed to fit your own needs.

The next input required is the amount of the check:

```
Amount of check ? 12.33
```

The computer then proceeds with the following results:

```
6-10-1983 $12.33
Input 99 if complete or new value? 99 <CR>
```

The date information above is obtained from the date used when you first started the system. If you wish a group of checks to carry a given date, you must initialize your system with that date.

The next question reminds you to set up your printer for printing checks:

```
Check if printer and forms ready. <CR> when complete.
```

The computer now proceeds with the printing of checks as follows:

```
Internal Bytes Management, Inc.
Accounts Payable
123 Solid State Road
Electron, AL lllll
lll-lll-llll
                                06-10-1983                    $123.44
Anderson
Sara and Rower
123 Beller Blvd
Retailville, AL lllll
Internal Bytes Management, Inc.
Supplier's No.              Invoice Date              Invoice Amount
llll                         06-10-1983               $123.44
```

As you can see, a form is needed to complete the information. A legal check must say "pay-to-the-order-of," have the clearinghouse number of the bank being used, and have the magnetic code used by banks to sort and route checks. All the remaining information is included.

After the check has been printed the computer returns to the program menu to wait for additional instructions.

## 9.2 Using The Supplier's File System

For this example run, we will examine use of the Supplier's file logic. The purpose of the Supplier's File option is to either produce checks to suppliers from the supplier's diskette data file or, to perform file maintenance on the

supplier's file. To examine this option begin from the beginning of the program by typing RUN from the keyboard and all prompts as before until you come to the following prompt:

```
Select output type:
     (F)ile update
     (C)heck
     (E)nd
     ? F <CR>
```

This time select the file option to illustrate the supplier's file system used. The next question is:

```
                    File Maintenance Menu
(A)dd new supplier
(D)ELETE a file
(E)nd cycle
(S)tart a new file
(U)pdate an existing file
? S <CR>
```

You must use the startup routine to initialize the end of file marker in your system. If you do not, your system will not work. If you use the startup routine a second time, it will be impossible for the system to operate. Be very careful when using this routine. The S option will erase any previous supplier file and setup file no. 1 with the word "END" in all string fields and "9999999999" in all numeric fields.

The overall design of the file maintenance system is the same as the customer file system. Since we have already detailed the operation of that system, we will not do so again here. You are referred to Chapter 7 for the details on how the supplier's system works. The difference between the two systems is the headings and information contained.

## 9.3 Adding Keyboard Input

The program may be used with all input generated from the keyboard, or some information so generated and the rest obtained from the file, or all of the information obtained from data files. In the combined approach, the information obtained from the file is that carried in the supplier's file. It is not necessary to input the name and address of a given supplier over and over again. You only need ask the computer to search the files. You may search on any of the fields except the amount due. To find a particular file, you need know only one of the various pieces of information.

The program starts in the same manner as when all input is from the keyboard. The beginning questions and answers are:

```
**** CHECK WRITING PROGRAM ****
                by
      Steven M. Zimmerman, Ph.D.
            Leo M. Conrad
      Donald R. Smith Ph.D., P.E.
        (Press <CR> to Begin)?
```

```
Disk Drive for data files (A,B,C,D)? B <CR>
Transaction file
     (I)nput
     (O)utput
     (N)one ? O <CR>
Name of file? CHECK.MAY <CR>
Select output type:
     (F)ile
     (C)heck
     (E)nd
     ? C <CR>
New starting number (Y/N)? N <CR>
```

You must use disk file input to post transactions to individual accounts.

```
Select input method--keys or disk (K/D)? D <CR>
```

At this point, rather than being asked to input a supplier's name and address, you are asked to find the organization as follows:

```
                         File Input Menu
-1     File Number
 0     Supplier's Name
 1     Company's Name
 2     Number And Street
 3     City
 4     State
 5     Zipcode
 6     First Name
 7     Title? -1
File - Supplier's Number? 1 <CR>
```

The computer now responds with the information in the first file.

```
Number              Title                          Input
0                   Supplier's Name                Anderson
1                   Company's Name                 Sara and Rower
2                   Number and Street              123 Beller Blvd
3                   City                           Retailville
4                   State                          AL
5                   Zipcode                        11111
6                   First Name                     Sam
7                   Title                          President
8                   Balance Due                    12.33
99 to continue, 98 to search more, or number to change? 99 <CR>
```

You may correct any information in the file, use the 98 option to continue to search the file, or use the 99 option to continue with the program. Assume you selected the 99 option to continue with the program. The next questions you will see are those associated with checkwriting.

```
Input invoice number? 22222
Purchase order number? 33333
Amount of check ? 123.44
01-10-1982     $123.44
Input 99 if complete or new value? 99 <CR>
Check if printer and forms are ready. <CR> when complete? <CR>
```

The checks are now printed.

## 9.4 Using Transaction File Input

When transaction file input is used, there are fewer questions than above. Assume you have prepared the following file:

| No Trans No | Date | DESCRIPTION | ACCOUNT DEBIT | CREDIT | AMOUNT DEBIT | CREDIT | St |
|---|---|---|---|---|---|---|---|
| 1 221 | 07-10-198 | Sara and Row | 2002 | 1 | 123.44 | 123.44 | IS |
| 2 222 | 07-10-198 | Franker and | 2003 | 1 | 44.45 | 44.45 | IS |

The above file was printed in the TRANSaction program. It was created in the checkwriting routine as the output of one of the manual systems. The input and output files produced and used by the checkwriting routine are exactly the same. The key difference is that you can create a file that posts to 2001 the overall accounts payable number, while if you are going to use the file as input, it must post against individual accounts. Since 2001 is the overall number, 2002 is the first supplier, 2003 is the second supplier, and so forth.

Input information needed when using a transaction file for input is:

```
**** CHECK WRITING PROGRAM ****
                by
      Steven M. Zimmerman, Ph.D.
            Leo M. Conrad
      Donald R. Smith Ph.D., P.E.
        (Press <CR> to Begin)?
```

After the preliminary prompts, the following prompts are asked:

```
Disk Drive for data files (A,B,B,C)? B <CR>
Transaction file
     (I)nput
     (O)utput
     (N)one ? I <CR>
Name of file? CHECK.MAY <CR>
```

At this point the checks are produced.

## 9.5 Examining The Program

The program is identified in line 10, and the remarks in lines 20 and 30 identify the files used by it. In lines 40 and 50 are some formats needed for print using statements later in the program. In line 60, the screen is cleared and the number of string variables and double-precision variables in the supplier's file are identified.

In Section One of the program, starting in line 70 and continuing through line 90 are the headings used for supplier's file input and output, while in lines 100 through 120 the headings are printed. The first question about the disk drive used appears in line 120. Also, in line 20, note the GOSUB 640. This statement directs the program to the subroutine that opens the supplier's file. This file remains open during the execution of the entire program.

The input/output option question is asked in line 130, while the name of the transaction file is input in line 140 where it is combined with the disk information to form a name that may be used to open disk files. In line 150, the file is opened for either input or output, depending on the option selected.

The heading is printed for a second time in lines 160 and 170 followed by the output menu that is printed in lines 180 and 190. If the option of ending the program is selected in line 190, then line 200 closes all files after printing the trailing data line information as required.

Line 210 directs the flow of the program to line 670, if file update is called for, while line 220 asks whether a new starting number is needed. If the answer is no, the program in line 230 opens x:NOCHECK.DAT to input the same information from a disk file. If the answer is yes, then line 240 inputs the information from the keyboard.

The name of your company must be placed in a disk file in the startup routine. This program calls in this file to obtain the information in line 250. If you have not used the startup program to perform this task, you will not be able to proceed.

If file input is being used, line 260 directs the program to line 1150. The keyboard input needed for transaction file input operation is complete.

Line 270 is the input menu, and lines 280 through 300 direct the action of the program according to the type of input selected. Lines 310 through 370 print the information about the supplier and allow corrections to be made. This same code is used no matter what routine is being performed.

The invoice number is input in line 390, and the purchase order number is input in line 390. Note: INVNUM is the invoice number, and PO$ is the purchase order number. You may use string variables for the purchase order number. You also may wish to substitute one for the other in the printing of the check depending on your particular needs.

In line 400, the amount of the check is input. The date is printed in line 410. The information is printed and the opportunity to make corrections occurs in lines 430 to 450. The "if-printer-is-ready,-hit-ENTER" question is in line 460. If you wish to remove this question, change line 460 to read as follows:

```
460 DAT$=DATE$:GOSUB 1240:GOTO 160
```

Section II of the program goes from line 470 through 610. This section is designed to input information from the supplier's file. If you are using the input values from keyboard and input supplier's information from file, this section is used. Note: The main menu for this section is found in lines 500 and 510.

Section III starts with the subroutine that opens the supplier's file in lines 630 through 660. If you are making any changes to the file structure, you must be very careful that you understand how the file is set up. The remarks in this section are designed to help you with this task.

The next part of this section performs file maintenance for the supplier's file. The main menu for this task is in lines 670 through 700, and the routing code is in lines 710 through 760. Line 770 starts the add option. Line 820 starts the delete and update options. Line 990 starts the startup option.

In line 1020 is the routine that allows you to search for a customer who owes you a specified amount of money. This routine continues through 1090. Lines 1100 through 1130 are part of the routine that gets information out of the random disk file. This code takes information out of the file buffer before it is destroyed by some additional use of the buffer.

Lines 1140 through 1220 are used to read in the transaction file to print checks. The program is designed to verify that the transaction numbers are within the limits defined by the number system being used.

The actual printing of the check is performed next. To move the print information more to the left, lower the variable LEFT in line 1240. Increase its value to move it more to the right. This will help you line up the information to print checks on your printer. The LPRINT lines 1270 through 1320 may be replaced by LPRINT statements without information, if the checks you are using have your company's name and address printed on them.

The number of LPRINTs in line 1330 may be varied, depending on the spacing needed for the checks you are using. Note: This system is configured to print the checks first, followed by the statement.

INVNUM is printed in line 1420. If you wish to change this value, as reviewed earlier, just substitute the variable PO$ for INVNUM.

The posting routine for transaction is in lines 1470 through 1570. There are REMark statements in lines 1580 and 1590 that explain what is being done. The file lines of code, lines 1600 and 1610, are used to reduce rounding errors that occur in all computers.

## 9.6 Summary

We have reviewed the operation of the program. For those interested in using the program only, this is the only part of the chapter you need to review in detail. We have reviewed the code also and if you wish to customize the program, this detail will help you to identify where and how the various program parts perform their respective tasks.

# 10. Using TRANSaction To Balance Your Cash Account

One utility, called (B)alance in the program menu, was added to the TRANSaction program to help you reconcile your cash account with the bank. With this utility, all you do to perform a reconciliation is to prepare a file of all transactions that have not cleared the bank account. This can be performed by one of the options in the balance routine. Then prepare a second file of adjustments. Adjustments are defined as transactions that have cleared the bank but were not keyed into the transaction file before the accounting reports were prepared, for example, nonrecorded checks and deposits.

Once the two files are prepared, they are merged together in the transaction program and run through the balance printing routine.

## 10.1 Running The Program

Refer to the TRANSaction program (Chapter 6) on how to prepare a transaction file and how to use the (K)lear option to tell the computer which transactions have been cleared and which are still outstanding. In this option, the ability to mark a transaction CLEAR, ISSUED, or ADJUSTMENT was available. The option should be used with the transactions of adjustment file, so mark this file for reference purposes. Note: The file of adjustments should be prepared as if it were to be posted in the normal manner, as it will be the next time period. (The objective of organizing the routine in this manner is to minimize keying.)

When the balance option is selected, you will see the following instruction and menu:

```
The objective of this routine is to balance your cash account with
your bank. It is assumed you have created a future action file,
have recorded your current period file, identified the cleared
transactions in your current period file, and have an old not
cleared file.
    (D)elete cleared transactions, (P)rint results?
```

The first option goes through your current period file and prints (to a disk file you will name next) a file of all uncleared accounts. The next question you will see when you select this option is:

```
Name of file with uncleared transactions?
```

After writing the new file on the disk, you still have the original file with no changes in it. You should make sure it is recorded and then exit the program to start over again with the not cleared file as your base.

The adjustment file we spoke of earlier is merged to the not cleared file.

Then you may return to the balance option and select P for printing the results. When you do, you will be asked the following questions:

```
Input identification of run?
Number of cash account?
Ending balance on balance sheet?
```

The computer now will take over and determine if there is a difference between the balance shown by your bank and your cash account. If there is, it is your task to trace the errors that may have occurred. In a computer procedure such as this, the most common error is an entry error.

This program subtracts all credit cash transactions that are adjustments, adds all cash debit transactions that are adjustments, adds all credit cash transactions that are issued but not cleared, and subtracts all debit cash transactions that have not cleared. The procedure is not the normal one used in a trial balance or the balance sheet.

## 10.2 Summary

This method of using the transaction program allows for the reconciliation of your cash account with the bank with a minimum of effort. The transactions keyed for this task are ready to use the next time period, except for the code associated with the adjustment. This procedure should help reduce the amount of time you spend on your checkbook. It does not perform the task of finding and eliminating errors, however.

# III: EVALPAC

# 11. Statistical and Analysis Menu Including 12 Business Routines

This chapter is a discussion of the EVALPAC menu. It is a collection of fourteen business routines that are useful in the day-to-day practice of business. The collection is integrated into a single package for ease of use and availability.

There are two objectives to the package. The first is to provide the business person with programs that are of value in accounting and other activities. The second is to provide business analysts with a collection of programs with which almost any business situation can be studied. The collection includes time value of money techniques, break-even analysis, and payback.

The (*)Cash flow analysis for real estate investments is designed to help the investor determine whether a given investment is economically feasible. The program has an integrated package of evaluation tools designed for this single purpose.

Regression analysis represents one of the most often used applied statistical analysis techniques in business applications. This program is written to perform regression and correlation analysis with the aid of both low- and high-resolution graphics. The regression line, confidence limits, and prediction limits are all plotted by the program, which is most useful, because it gives a visual picture of the forecasting problem.

Since an Apple CP/M diskette can hold approximately 127Kb of programs and/or data, Cash and Forecast programs are not a part of EVALPAC code but are shown on the EVALPAC menu. When selected from the menu, EVALPAC loads either Cash or Forecast in place of EVALPAC, which "wipes out" the program and begins running whichever of the two programs you specified. Thus, EVALPAC is a menu program for Cash and Forecast. Cash, or Forecast can be referenced separately from CP/M by entering:

```
>A MBASIC CASH <CR>
or,
>A MBASIC FORECAST <CR>.
```

## 11.1 What's In The EVALPAC Package?

The following programs are included in this package:

1. An amortization program with the capability of handling equal payment loans, equal reduction loans, and interest only loans.
2. Programs to handle all five types of depreciation now being used: straight line depreciation, sum of the year's digits, declining balance, and the procedures as outlined in the 1981 tax reform act.
3. A break-even point program to plot break-even charts and do a payback period analysis.
4. A program to calculate the future value of an asset.
5. A program to calculate the present discounted value of a series of unequal annual cash flows.

6. A program to calculate the discounted value of a bond or a mortgage with or without a balloon call option.
7. A program to calculate any of the missing factors associated with a mortgage.
8. A program to calculate any of the missing factors of a mortgage that has a call balloon payment.
9. A Cash flow program.
10. A regression and forecasting program.

The program collection is designed to operate from a menu that allows the user to select the specific routine of interest.

## 11.2 Organization Of This Section

The section is divided into sixteen chapters, one for each routine in EVALPAC and one for each of the other programs in the menu. In each chapter, the general concept of the program is introduced, the operation procedure is detailed, some theory is presented (where applicable), and the code associated with the routine is examined. In this section, the operation of the main menu will be detailed along with the code associated with this part of the program.

## 11.3 Running EVALPAC

After selecting EVALPAC by typing A <CR>, you will see:

```
**** EVALPAC APPLE CP/M PROGRAM ****
                by
        Steven M. Zimmerman, Ph.D.
              Leo M. Conrad
        Donald R. Smith, Ph.D., P.E.
          Press <CR> to Begin ?
```

Simply press the <CR> key to continue to the EVALPAC menu:

```
                *** EVALPAC MAIN MENU ***
(A)MORTIZATION OF LOAN
(B)REAK-EVEN ANALYSIS
(C)ALL BALLOON NOTE VALUE
(D)ECLINING BALANCE DEPRECIATION
(E)ND
(F)UTURE VALUE OF AN ASSET
(M)ORTGAGE CALCULATIONS
(P)RESENT VALUE OF UNEQUAL CASH FLOWS
(R)EAL ESTATE PROPERTY 1981 LAW DEPRECIATION
(S)TRAIGHT LINE DEPRECIATION
(V)ALUE - DISCOUNTED OF A BOND/MORTGAGE
(X)DEPRECIATION PERSONAL PROPERTY 1982 CHANGE
(Y)EARS DIGITS-SUM OF -- DEPRECIATION
(*)CASH FLOW ANALYSIS FOR REAL ESTATE INVESTMENT
(#)FORECAST Regression and Correlation Analysis
*** ?Enter Your Choice- ?
```

You will return to this menu after completion of each routine. The operation of the menu is simple. Select the desired option by typing the letter in the brackets and pressing <CR>. Even if you type additional material on the line, the computer looks at the first character only. The computer will accept both lower and uppercase letters. An "a" or an "A" both result in the amortization routine being called to the screen.

If your selection is not within the set of options, you will be cycled back to the menu. The program logic is designed to minimize certain data input errors.

## 11.4 Chapters, Routines, And Programs

Listed below are the chapters where you will find the various routines from EVALPAC.

| Chapter | Routine |
|---|---|
| 12 | (A)MORTIZATION OF LOAN |
| 13 | (B)REAK-EVEN ANALYSIS |
| 14 | (C)ALL BALLOON NOTE VALUE |
| 15 | (D)ECLINING BALANCE DEPRECIATION |
| 16 | (F)UTURE VALUE OF AN ASSET |
| 17 | (M)ORTGAGE CALCULATIONS |
| 18 | (P)RESENT VALUE OF UNEQUAL CASH FLOWS |
| 19 | (R)EAL ESTATE PROPERTY 1981 LAW DEPRECIATION |
| 20 | (S)TRAIGHT LINE DEPRECIATION |
| 21 | (V)ALUE - DISCOUNTED OF A BOND/MORTGAGE |
| 22 | (X)DEPRECIATION PERSONAL PROPERTY 1981 LAW |
| 23 | (+)SUM OF THE YEARS DIGITS DEPRECIATION |
| 24 | (*)CASH FLOW ANALYSIS FOR REAL ESTATE INVESTMENT |
| 25 | (#)FORECAST Regression, and Correlation Analysis |

## 11.5 Examining EVALPAC

The program is listed in Appendix A at the end of this section. All discussions in this section will refer to this listing. The credits and menu start in line 10 and continue through line 400. Line 10 is a REMark statement to identify the -name of the program and help the programmer or individual copying the program save it correctly. If you wish to copy this program, you can save the most current version by listing line 10. Then delete the REM and press <CR>. This helps reduce errors in recording the program while it is under development.

Line 20 clears the screen and clears 500 bytes of memory for string variables. Line 30 sets the dimension limitations for some routines included in the program. We will refer to this line when the specific routine is discussed. Line 40 establishes a format to be used for printing dollar and cents information.

Lines 60 through 110 print the credits for the program. Lines 120 through 260 are the main menu. Line 120 is the line to which all routines return after they have completed their work.

Lines 270 through 400 route the flow of the program depending on the answer given in the menu. In all cases, the LEFT$ function is used to pick the first character sent for examination only. The program looks for both lower and uppercase letters to satisfy the question.

Move now to Chapter 12 for the beginning of the EVALPAC series of programs and a venture into forecasting and evaluation techniques utilizing the power of Apple CP/M.

# 12. LOAN (A)MORTIZATION

Almost every business arrangement involves some use of borrowed funds. This routine helps identify the meaning of the various ways financing can be done. The three classic methods of borrowing funds, equal payment, equal reduction, and interest only, are all handled by this routine.

## 12.1 Running The Program

(A)MORTIZATION OF LOAN is selected from the main menu by typing A. The first thing you will see is:

```
                    *** OUTPUT MENU ***
(C)RT
(P)rinter
Enter your choice C/P ?
```

This menu allows you the opportunity to obtain a hard copy of the analysis or view the analysis from the screen. Assuming you desire screen output, press **C** <CR> and you will see:

```
BALLOON (Y/N)? Y <CR>
```

During times of difficult financing, a call or "balloon" of the principal due is not unusual. A typical example may be when payments are calculated assuming a 25-year payout, and after five years the remaining balance becomes due and payable. Notes of this type give investors a financial advantage in the call for cash during the life of the note, usually at the expense of a very large payment when the call is due.

In this example, it is assumed that a balloon payment is part of the loan agreement. The next question is:

```
PAYMENT # OF BALLOON ? 12 <CR>
```

Since a balloon may be included in a contract at any time, not necessarily at the end of an investment year, the payment number is needed to perform the calculations. It is assumed the payment schedule for call is at period 12.

The next question is:

```
MONTH NUMBER IN WHICH PAYMENT STARTS? 7 <CR>
```

This amortization program is designed to sum the amount of interest paid (or collected) per year and the amount of the reduction of the loan. This ability makes the calculation of taxes a little easier. For the program to perform this task, it must know the month in which the first payment is made. In this case, month 7 is July, and there are 6 payments to be made the first year (July, August, September, October, November, and December).

The next question is the payout menu:

```
PAYOUT MENU
     EQUAL (P)AYMENT
     EQUAL (R)EDUCTION
     (I)NTEREST ONLY ? P
```

Many types of loans exist. The most common are the three listed above. In an equal payment loan, the same amount is paid every month or period except the last. The last month is usually different to make up for rounding errors that occur due to interest variances in this type calculation.

In an equal reduction loan, the payments each period vary, but the amount of loan reduction is the same. A common loan of this type is to borrow 1000 for 10 years. Each year the loan is reduced by 100 and interest is paid on the outstanding amount for the period.

Interest only loans consist of loans where the principal is given the lender. Interest payments are made each period. At the end of the life of the loan, the principal and interest for the last period is due.

Warning! Academic definitions for interest terms and those used by your local banker may not be the same. The academic definition of simple interest loans is the interest only type loan. Current banking use refers to simple interest as interest on the remaining balance to be included in the payment each period. This then is actually an equal payments type of loan.

For this part of our discussion, P (equal payment) was selected. The next set of input prompts (presented in a tabular format) are:

| $Amt.of Loan | APR(%) | No.Pay/Yr | Amt.of Payment($) (0) to Cal. | No. of Yrs. |
|---|---|---|---|---|
| ?1000 <CR> <CR> | 12 <CR> | 12 <CR> | 0 <CR> | 5 |

The first item in this set of questions was how much is the value of the loan. The answer was $1000. The next question was annual percentage rate (APR). The answer was 12, meaning 12 percent. The program is designed to convert the answer to decimal format to be used in the calculations. Be careful how you input this type of data!

The next item was payments per year. To maximize the flexibIlity of the program, monthly payments are not assumed, but the user may specify the payments per year. It is assumed the number of payments will be one or greater, and that an integer number of payments will be made per year. The program cannot handle a value such as 6.5 payments per year.

The amortization program used the mortgage calculation routine to calculate the payment per period for this type of loan. If the payment is specified from another source, you may input it here. If you input 0, the computer will do the calculations for you. You can expect some differences between the numbers as calculated by various computers and calculators. These differences come from rounding errors that are a constant problem in this type of work.

The last item was the number of years. The program expects an integer

number of years but usually will work if you input some other value, for example, 2.5 years for a 30-month loan. You may experience some rounding trouble when looking for an 8 month loan, .6666667 years. Double check all loans of this type to make sure you did not input one extra or one too few periods.

After you have completed the above, the program logic produces the following:

```
LOAN=      1000    APR=    .11999999731779    PAYMENTS PER YEAR= 12
YEARS=      5      PERIOD PAYMENTS= 22.24
PERIOD             INTEREST      REDUCTION     LOAN          PAYMENT
<CR> TO PAGE ?
```

When producing output on the monitor, the program is designed to stop periodically so the user can retrieve information from the screen. Notice that the information matches the data you typed in, except the APR. This value suffers from the rounding that occurs internally in the computer. You can see where some of the rounding errors are generated.

After you press <CR>, the computer will produce the following:

```
 1          10.00          12.24          987.76          22.24
 2           9.88          12.36          975.40          22.24
 3           9.75          12.49          962.91          22.24
 4           9.63          12.61          950.30          22.24
 5           9.50          12.74          937.56          22.24
 6           9.38          12.86          924.70          22.24
 0          58.14          75.30          924.70         133.44
<CR> TO PAGE ? <CR>
 7           9.25          12.99          911.71          22.24
 8           9.12          13.12          898.59          22.24
 9           8.99          13.25          885.34          22.24
10           8.85          13.39          871.95          22.24
11           8.72          13.52          858.43          22.24
12           8.58         858.43            0.00         867.01
 0          53.51         924.70            0.00         978.21
```

After you press the <CR> key as your answer to the above question, you will see the main menu. Before proceeding, look at the last line of output. The "0" line produces the totals for each column.

Below is the list of questions and answers for an equal payment example where there is no balloon payment:

```
OUTPUT MENU
     (C)RT
     (P)rinter ? C
BALLOON (Y/N) ? N
MONTH NUMBER IN WHICH PAYMENT STARTS ? 11
PAYOUT MENU
     EQUAL (P)AYMENT
     EQUAL (R)EDUCTION
     (I)NTEREST ONLY ? P <CR>
```

```
$Amt.of Loan   APR(%)        No.Pay/Yr          Amt.of Payment($)   No.of Yrs
                                                    (0) to Cal.
1000 <CR>      12 <CR>       1 <CR>             0 <CR>              5 <CR>
LOAN=   1000   APR=.119999999731779
YEARS=    5    PERIOD PAYMENTS=277.41                    PAYMENTS PER YEAR=1
   PERIOD          INTEREST        REDUCTION                 LOAN      PAYMENT
<CR> TO PAGE?
1                    120.00           157.41               842.59       277.41
2                    101.11           176.30               666.29       277.41
3                     79.95           197.46               468.83       277.41
4                     56.26           221.15               247.68       277.41
5                     29.72           247.68                 0.00       277.40
0                    387.04         1,000.00                 0.00     1,387.04
<CR> TO PAGE?
```

Notice the last payment for year five is one cent less than all the other payments. This is the adjustment needed to correct for rounding errors in interest calculations. In this last example, the results were not summed each year, since it was a yearly payment situation.

The next example is for an equal reduction loan. Look at the questions carefully to see how they differ after the loan type is declared:

```
OUTPUT MENU
  (C)RT
  (P)rinter ? C <CR>
BALLOON (Y/N) ? N <CR>
MONTH NUMBER IN WHICH PAYMENT STARTS? 1 <CR>
PAYOUT MENU
  EQUAL (P)AYMENT
  EQUAL (R)EDUCTION
  (I)NTEREST ONLY ? R <CR>
```

```
$ Amt.of Loan    APR(%)          No.Pay/Yr   Reduction   No. of Yrs.
1000 <CR>        12 <CR>         2 <CR>      100 <CR>    5 <CR>
LOAN= 1000         APR= .119999999731779
YEARS=    5          PERIOD PAYMENTS= 0          PAYMENTS PER YEAR=2
PERIOD               INTEREST     REDUCTION        LOAN        PAYMENT
<CR> TO PAGE? <CR>
 1                     60.00         100.00       900.00         160.00
 2                     54.00         100.00       800.00         154.00
 0                    114.00         200.00       800.00         314.00
<CR> TO PAGE? <CR>
 3                     48.00         100.00       700.00         148.00
 4                     42.00         100.00       600.00         142.00
 0                     90.00         200.00       600.00         290.00
<CR> TO PAGE? <CR>
 5                     36.00         100.00       500.00         136.00
 6                     30.00         100.00       400.00         130.00
<CR> TO PAGE? <CR>
 7                     24.00         100.00       300.00         124.00
 8                     18.00         100.00       200.00         118.00
<CR> TO PAGE? <CR>
 9                     12.00         100.00       100.00         112.00
10                      6.00         100.00         0.00         106.00
 0                     18.00         200.00         0.00         218.00
<CR> TO PAGE? <CR>
```

In this case, the reduction of loan was a constant $100 per period while the payment varied from $160 to $106. Be careful when using this part of the amortization routine. If you reduce the amount of loan to a negative value, the computer will keep producing results as requested. A five year loan for $1,000 reduced by $300 per year makes no more sense to the computer than it does to a human being.

In our last example, the questions and answers for an interest only loan are examined:

```
OUTPUT MENU
  (C)RT
  (P)rinter ? P <CR>
  BALLOON (Y/N) ? Y <CR>
MONTH NUMBER IN WHICH PAYMENT STARTS? 1 <CR>
PAYOUT MENU
  EQUAL (P)AYMENT
  EQUAL (R)EDUCTION
  (I)NTEREST ONLY ? I <CR>
```

| $Amt.of Loan | APR(%) | No.Pay/Yr | Reduction | No. of Yrs. |
|---|---|---|---|---|
| 1000 <CR> | 12 <CR> | 12 <CR> | | 4 <CR> |

```
TITLE, DATE (XX/XX/XX)? Sample run,08/08/82
LOAN=  1000     APR=.11999999731779
YEARS=   4      PERIOD PAYMENTS= 0          PAYMENTS PER YEAR= 12
PERIOD          INTEREST     REDUCTION     LOAN          PAYMENT
<CR> TO PAGE?
```

The following will be produced on your printer:

```
Sample run,                08/08/82
 LOAN=  1000        APR=.11999999731779        PAYMENTS PER YEAR=12
  YEARS=   4         PERIOD PAYMENTS= 0
    PERIOD            INTEREST       REDUCTION        LOAN      PAYMENT
       1                10.00            0.00      1,000.00      10.00
       2                10.00            0.00      1,000.00      10.00
       3                10.00            0.00      1,000.00      10.00
       4                10.00            0.00      1,000.00      10.00
       5                10.00            0.00      1,000.00      10.00
       6                10.00            0.00      1,000.00      10.00
       7                10.00            0.00      1,000.00      10.00
       8                10.00            0.00      1,000.00      10.00
       9                10.00            0.00      1,000.00      10.00
      10                10.00            0.00      1,000.00      10.00
      11                10.00            0.00      1,000.00      10.00
      12                10.00        1,000.00          0.00   1,010.00
       0               120.00        1,000.00          0.00   1,120.00
```

## 12.2 Theory

The theoretical base on which the three alternate methods of financing are designed depends on how interest is calculated and paid. The ideas are simple, but the mathematics can get complex in some situations.

In an equal payment loan, the interest is calculated per payment period based on the amount of the loan at the beginning of the period. The interest charged per period is determined by dividing the annual percentage rate by the number of periods per year.

For example, if the annual rate is twelve percent, then the monthly rate charged for monthly payments would be one percent (twelve divided by twelve). This is referred to as interest on the remaining balance.

The method used to find the payment that results in the loan being paid off in the time specified is an equation that can be found in any finance book. This equation is used in the program to find the payment, and then that number is used in the calculations.

Each period, the interest due for the month is calculated. This value is subtracted from the payment to determine the reduction of loan for the month. The new value of the loan is determined by subtracting the reduction of loan from the original balance. For the next month, the interest per period is multiplied by the new original balance, and the process repeats itself.

For equal reduction loans, the value of the loan is divided by the number of payment periods the loan will remain outstanding. This loan often involves single annual payments. The interest is calculated by multiplying the interest rate for the period by the original balance as in the case of an equal payment loan. The interest expense is added to the reduction of loan to determine the payment. For this type of loan, the payments will vary from period to period.

In an interest only loan, interest is paid each year (or other period) during the life of the loan. At the end of the loan, the interest for the last period is paid plus the original loan.

## 12.3 Examining The Program

Line 290 in the menu portion of the program directs the computer to line 780 when use of the amortization routine is selected. The first line in this routine is a REMark statement to identify the program. Line 790 defines the format to be used to output the results, while line 800 starts the input of data from the keyboard. Lines 810 through 830 are used to ensure that the answer given to the question in line 800 is an acceptable answer.

Lines 840 through 870 input the remaining information through the time of loan for the routine. In line 880, the routing process begins. In line 890, if you are not interested in an equal payment loan, you are routed around the code in lines 900 through 1020. In this last series of lines, all the calculations are performed for the equal payment situation. The equal reduction calculations are performed in lines 1150 through 1120. At this point in the routine, the program goes to line 1000 to take advantage of code already available for calculating the periodic charges.

In line 1130, the calculations for the interest only loan start. This routine continues through line 1220, when it also is routed to line 1000 for the periodic calculations.

Lines 1230 through 1380 are a series of subroutines that aid in the output of material on both the monitor and the printer. This completes the code needed for the amortization routine.

# 13. BREAK-EVEN Analysis

A simple yet powerful business analysis technique is the break-even analysis of a potential or existing business venture. The reason break-even analysis is so powerful is that it is a graphical technique. It usually is used to draw a picture of the investment that aids greatly in understanding the options available.

The limitations of break-even analysis are a direct result of the simplicity of the technique. In break-even analysis, only two types of costs are considered: fixed and variable. Fixed costs do not change as the level of product or sales changes, and variable costs change directly with sales. In real life, no costs are totally fixed, and none are purely variable. The second assumption on which break-even analysis is based is made for those analyses involving revenue and costs. In this type of problem, the user must assume that sales and production are closely related. This is seldom the case except in very elementary situations.

A person who is not well-versed in the details of financial analysis may better visualize the potential of an investment with a break-even analysis We recommend that the technique be used as a first approximation to determine whether a proposed venture should be looked into in more depth. If the production in pieces scale is divided by the output or sales per time period, the break-even chart can be converted directly to a payback period analysis. In many cases, this is a very useful conversion. This capability is part of our program. You get both a payback period and break-even analysis when you run the program.

## 13.1 Running The Program

The program starts as follows:

```
                    *** Break-Even Analysis ***
FIXED COST FOR ALT. NO. 1: ?
VARIABLE COST FOR ALT. NO. 1: ?
```

This set of prompts refers to the cost parameters of alternative 1 first. To illustrate, assume you are considering investing in a boat to rent. You already have made your decision to go into the business. The question you are concerned with is which of two boats you should purchase. The first boat costs $2,000 and has an operating expense of $11 per average trip. The second boat costs $1,000 and has an operating expense of $22 per trip.

For the example problem you would have the following:

```
FIXED COST FOR ALT. NO. 1: ? 2000 <CR>
VARIABLE COST FOR ALT. NO. 1: ? 11 <CR>
```

The next prompt set is a repeat of the first set, this time for the second alternative. In this case, answer as follows:

```
FIXED COST FOR ALT. NO. 2: ? 1000 <CR>
VARIABLE COST FOR ALT. NO. 2: 22 <CR>
```

Note: The fixed cost of alternative one was higher than that of two, while the variable cost of one was less than that of two. If this type of relationship does not exist, one alternative is always better than the other, and the program will tell you this situation exists. The program will not be able to plot the results under these conditions.

In the example, assume the decision to go into business has been made. If you want to compare a revenue function with a cost function, the program will do so. In this case, put the revenue per unit into the program as the variable cost for one of the alternatives. For that alternative there is no fixed cost, so input the fixed cost as zero.

The next question is:

```
PRODUCTION/SALES PER YEAR ? 1500
```

Assuming you would sell 1,500 units per year, the input data is complete and the program solves the problem and plots the results. The output will appear as shown in Figure 13.1.

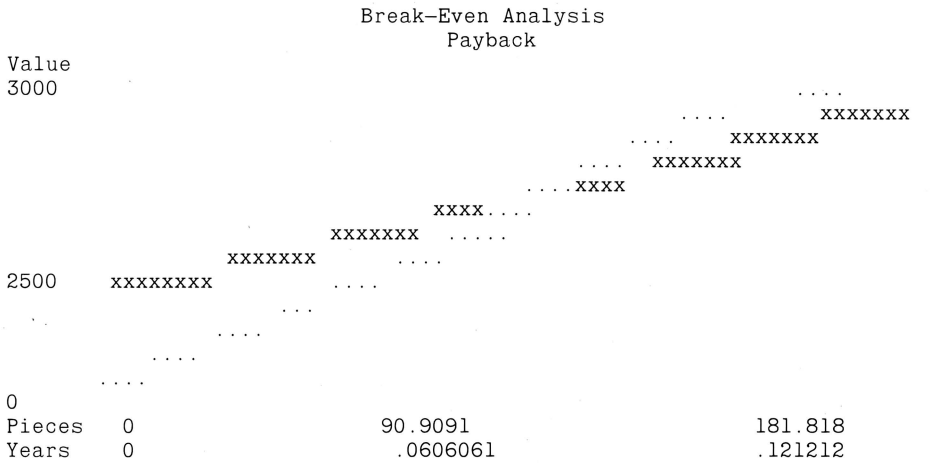



**Figure 13.1 Break-even Chart for the Example Problem**

## 13.2 Examining The Program

The program starts identification in line 3400 and continues through line 3780. In line 3410 you turn off the key display that usually appears on the bottom of the screen. Line 3430 clears the screen and sets the characters per line at 40 using the WIDTH statement. If you do not like the display in this manner, simply remove this statement.

In line 3440 are the credits followed by the first question in line 3450. The second question, fixed cost/variable cost, appears in line 3460. The last input question is in line 3470.

Lines 3480 through 3490 are the calculations needed to obtain the answers in this program. Line 3500 checks to see if the value of BREAK is negative. If it is, you are told of the results and returned to the beginning of the program.

Line 3510 begins the graphics part of the program with the SCREEN 1 instruction, and a change back to 80-character format. The headings for the program are printed in lines 3520 through 3580. The drawing of the box occurs in line 3590. The drawing of the lines that divide the box occurs in lines 3600 and 3610.

The actual program graphics are performed in lines 3620 through 3760. One problem may result in the lines not always going exactly through the center of the screen. The graphics are performed on an integer base, and rounding errors can occur in some circumstances.

In order to make this program as general as possible and not require the user to purchase additional special-feature graphics cards or software, we have attempted to design a simple (but slow) screen "graphics" display. If you specify the line printer as output (line 3520), then in line 3541 the POKE 3,150 places the decimal value 150 in memory location 3.

Memory location 3 for Apple CP/M is a special location. This location controls what is called the I/O Byte. The 150 value changes the bit configuration of this byte such that subsequent PRINT statements will be transferred to the line printer. However, the screen display is disabled. Thus, the output of the breakeven chart will be routed to the printer (assuming you have turned it on) and no screen display is produced.

The screen is enabled again in line 3765 by poking the value 149 into memory location three. The 149 value "toggles" the CRT back on. (When MBASIC is initially booted, the value of the I/O byte at memory location three for Apple CP/M is 149.)

Lines 3770 and 3780 allow you to look at the screen without interference until you are ready to return to the main menu.

## 13.3 Summary

Break-even analysis can give you a good approximation of the value of simple investments. Although the technique is simple, it does help in many investment situations. We recommend that you use it as a beginning analysis tool to help understand the relationships between cost functions or cost-revenue functions.

This program gives you a visual analysis of the break-even problem and shows the relationships being studied in a pictorial manner. Its value is further enhanced by the graphical presentation.

# 14. (C)all Balloon Note Value

This routine assumes you are concerned with an equal payment loan. The object is to calculate the value of the outstanding balance after a set number of periods. For an equal reduction loan the calculations would be very simple, since the amount of the loan is equal to the original balance, less the quantity of the number of payments, times the reduction per payment. In an interest only type loan, the amount outstanding is the original balance.

## 14.1 Running The Program

The program assumes you know the original value of the loan, the annual percentage rate, the life of the loan in years, the payments per year, and the payment number of the balloon payment. If you do not know one of the above factors but know the payment instead, it is necessary to use the (M)ortgage calculation routine (Chapter 17) before you use this routine. The questions asked and sample answers are shown below:

```
          ***** CALL BALLOON NOTE VALUE PROGRAM *****
PRESENT VALUE - INVESTMENT ? 1000 <CR>
APR ? 12 <CR> (This is the annual percentage rate as a %)
LIFE IN YEARS ? 4 <CR>
PAYMENTS PER YEAR ? 12 <CR>
PAYMENT # OF BALLOON PAYMENT ? 24 <CR>
```

The computer then generates the following results:

```
PAYMENT:           26.33
LOAN:       1,000.00
LIFE:  4
PAYMENTS/YEAR:  12
  12%
BALLOON NOTE          559.42
BALLOON PAYMENT # 24
<CR> TO CONTINUE?
```

The program is designed to reproduce the input data to help you eliminate input errors. There is no hard copy routine since everything comfortably fits on a single screen.

The above results mean that the balloon payment in 2 years (24 payments) will be $559.42, the payments will be $26.33 each, and the loan will be paid off in 4 years.

## 14.2 Theory

Two methods for calculating the value of a balloon note were considered. One method is to walk through the amortization schedule until the payment

is reached and examine what the remaining balance is. Even on a high speed computer, this is a very time consuming method of calculation. It is the most accurate method, because it takes into account the actual value of the payment and the rounding errors associated with the calculations.

The second method of calculating a balloon note is to find the present value of the remaining payments. This method does not take into account the difference in the last payment and is subject to some rounding error. It is the fastest and the error will be only a small percentage. This is the method selected for our program.

## 14.3 Examining The Program

In line 300, which is used to direct the flow of the program to the call routine, the variables SM$ and SL$ are defined as "C" so they may be used to direct the flow of the program at a later time. Line 300 directs the computer to line 500. Line 500 is in the middle of the (M)ortgage calculation routine. Double use is being made of this code.

Lines 500, 510, 520, and 530 are used to input the information about the loan being considered. Line 540 now directs the program to line 560 to calculate the payment associated with the loan. When the calculations are complete, the computer is directed from line 570 to line 1390.

The balloon note calculations are performed between line 1390 and line 1430. The program then is directed to line 710. In lines 710 through 770 the results are printed on the screen.

# 15. (D)eclining Balance Depreciation

The 1981 Federal Tax Reform Act dropped this method of depreciation and replaced it with a table lookup method called the Accelerated Cost Recovery System (ACRS). Many states have not yet seen fit to go along with this change, and the old methods of computing cost recovery (depreciation) are still in use. The idea behind this method of depreciation is that some ratio (1.25, 1.4, 1.75, 2) times the rate at which an asset is depreciated by the straight line method is a good representation of the depreciation pattern a concern should be allowed for a given asset.

Use of the Declining Balance method can present one problem to the user. Unlike the straight line and sum-of-the-year (SYD) methods, the DB method does not use the estimated salvage value of the asset. As such, the DB method of capital recovery possesses its own "built-in" salvage value. Normally, it is not possible to depreciate an asset down to a zero book value at the end of its depreciable life. The reason behind this is the fact that the DB method is computing a fixed percentage times a declining book value each year. Hence, it is not possible to reach a zero book value at the end of the asset's depreciable life.

## 15.1 Running The Program

After selecting the "D" option in the main menu, the first thing you will see on the monitor screen is:

```
*** DECLINING BALANCE DEPRECIATION ANLALYSIS ***
(C)RT, (P)rinter ?
```

Some assets that last a long time will result in listings too large to fit on the screen of your monitor, so the option of hard copy printing was included. When hard copy is selected, you will have the opportunity to input identification information to be printed on your printer.

Select "C" for the sample run, and the next thing you will see is:

```
YEAR OF PURCHASE ?
```

The program is designed to print the actual year rather than year 1, 2, .... This is a great help in future years by identifying where you are on a particular depreciation schedule. Use 1981 for the year this asset was first purchased.

The next question is:

```
DEPRECIABLE BASIS?
```

An asset is depreciated from its first cost to some salvage value. The first cost includes not only the purchase price but some of the installation costs and such. The term "basis" is used to refer to the amount of money that is to be recovered (depreciated), beginning at time 0. Your accountant can identify the costs that should be included. These costs are said to be capitalized.

For the example, use **$1,000** as the capital expense.

The next question is:

```
SALVAGE VALUE ?
```

The 1981 law allows businesses to depreciate assets to zero salvage value. Using the declining balance method, it is possible to input any salvage value zero or greater. Use **$100** for the example.

The next prompt is:

```
LIFE ?
```

The Internal Revenue Service and state tax offices use different guidelines for determining the life of an asset. The law is constantly changing. Check carefully the life you may use in each case. Use **6** for the sample problem.

```
MONTH OF PURCHASE ?
```

The law sometimes requires you to allocate depreciation in any given year to the nearest month, or the nearest six months, etc. This, like all other aspects of depreciation, is constantly being changed. Use **1** for this example.

The next prompt is:

```
RATE ?
```

Historically, the permissible "rates" for the DB method range from 1, 1.25, 1.5, 1.75, and 2.00. Most new personal property can be recovered using a DB rate of 2.00. Used property generally carries an assigned rate of 1.75. Elements of real property normally use an assigned rate of 1.50. For the example, assume a DB rate of **2**.

The results the computer will produce are:

```
YEAR          DEPRECIATION          CUMULATIVE          BOOK VALUE
1981                333.33              333.33              666.67
1982                222.22              555.56              444.44
1983                148.15              703.70              296.30
1984                 98.77              802.47              197.53
1985                 65.84              868.31              131.69
1986                 31.69              900.00              100.00
DO YOU WISH MONTHLY ALLOCATION (Y OR N)?
```

Since the item was purchased in month one according to the input, the allocation month by month should be identical to the above. The allocation was asked for with the following results:

```
YEAR          DEPRECIATION          CUMULATIVE          BOOK VALUE
1981                333.33              333.33              666.67
1982                222.22              555.56              444.44
1983                148.15              703.70              296.30
1984                 98.77              802.47              197.53
1985                 65.84              868.31              131.69
1986                 31.69              900.00              100.00
1987                  0.00              900.00              100.00
<CR> TO CONTINUE ?
```

When the monthly allocation is run, the depreciation usually is straightened out for one additional year. This was not the case in the example, but 1987 was printed, nevertheless.

## 15.2 Theory

The idea behind an accelerated depreciation method is that an asset loses its value faster when new than when it is older. This is generally true, but depreciation as used for tax purposes is purely a taxing method and may or may not have anything to do with the actual value of an asset.

To understand how the rate is a ratio associated with straight-line depreciation, assume you have an asset that will last 10 years. Using straight line depreciation, 1/10 of the value is depreciated each year. Using double declining balance, 2/10 of the remaining value may be depreciated each year. The key point is that no matter what method is used, the total amount of depreciation cannot exceed the original value of the asset less salvage value, if any.

## 15.3 Examining The Program

The declining balance code starts in line 390, where the title variable is named and the computer is transferred to line 1640. All three of the traditional techniques of depreciation share the use of the code in lines 1640 through 1760, defining title variables and inputting data. Line 1770 is executed for declining balance only, inputting the value of the RATE.

Lines 1780 through 1820 continue with defining initial values to variables and defining title variables. Line 1830 performs the task of inputting identification information for the printer output option, while line 1840 begins output of the identification information.

Lines 1850 through 1880 initialize some additional variables and define the format for output. Line 1890 starts the output by printing the headings for the columns of information produced by this program.

Lines 1910 and 1920 route the program to another code if other than the declining balance option was selected. Line 1930 begins the calculations for the declining balance depreciation method. The calculations continue through line 2060 until complete, or until the program is transferred to the straight line depreciation code in line 1990 or the termination sequence in line 1980.

The program automatically transfers to straight line depreciation whenever it is best to do so, based on which alternative yields the greatest depreciation. Line 1990 makes this decision.

In line 2030, the output subroutine in lines 2310 through 2350 is used to produce output by monitor and printer. This same subroutine is used for all three traditional methods of depreciation.

After the program is transferred to the straight line code beginning in

line 2170, it continues to produce the depreciation schedule until the end of the life of the asset. The straight line code runs from 2170 through 2190.

The allocation by month choice is made in line 2200. This code is shared by all other methods. The allocation output is produced in lines 2210 through 2300. The output subroutine in lines 2310 through 2350 is used to output from this part of the program.

# 16. (F)uture Value Of An Asset

One of the objectives of EVALPAC is to provide the user with a collection of tools capable of doing a complete financial analysis of any business situation. The need to calculate the future value of an asset arises in many business situations, for example, in forecasting the potential resale price of a real estate holding that appreciates with time at approximately the same rate as inflation.

## 16.1 Running The Program

After selecting the "F" option in the main menu, the first thing you will see is:

```
***** FUTURE VALUE OF A PRESENT SUM OF MONEY *****
PRESENT VALUE ?
```

The objective of EVALPAC is to provide the user with a set of tools capable of doing almost any financial analysis. Tools must include the ability to forecast the future value of an asset such as real estate due to inflation or the ability to find the future value of a sum of dollars placed in a bank earning some given effective interest rate.

Assume you have a piece of land valued at $1,000 and the inflation rate is ten percent. The value of the land has kept up with inflation for the past twenty years, so you assume it will continue to do so. You want to know the value of the land in five years.

In a similar manner, place $1,000 in a bank earning ten percent per year. You wish to know the amount of dollars in the bank after five years. This problem and the one above are the same.

Answer **1000** <**CR**> to the above question and you will see:

```
NUMBER OF PERIODS ?
```

The problem defined was for five periods. Enter **5** <**CR**>.

The result of the analysis is:

```
FUTURE VALUE: 1610.510110855103
<CR> TO CONTINUE ?
```

You may wonder why the number was allowed to run on and did not print the results to the nearest penny. You may wish to do your analysis in terms of millions of dollars rather than dollars and cents. If this is true, the above output would be of maximum value.

## 16.2 Theory

The idea behind this routine is simply the idea of compound interest when investing some fixed dollar amount. The future value is calculated based

on what the original value earns and on what the interest retained in the investment earns. The future value at the end of any period is equal to the value at the beginning of the period plus the interest earned for the period. This value (interest plus value at the beginning of the period) is the basis for calculating the next period.

## 16.3 Examining The Program

The program directs the flow of the activities from line 310 to line 1440 when the "F" option is selected. Line 1440 identifies the beginning of the program with a REMark statement. Lines 1450 through 1470 input the information needed by the routine.

Line 1480 does the calculations, and line 1490 prints the results. The remainder of the program directs the computer to return to the main menu after the user has obtained the results needed.

# 17. (M)ortgage Calculations

There are four factors involved with an equal payment mortgage. These factors are:

1. Periodic PAYMENT
2. INVESTMENT, original loan
3. Annual Percentage rate
4. The life of the loan in payments (Years and payments per year).

If you are given any three of the above facts, you then may compute the other. This routine is designed to perform this task.

## 17.1 Running The Program

The first question you will be asked is:

```
***** MORTGAGE CALCULATION PROGRAM *****
MORTGAGE MENU Solve for the following
     (P)AYMENT
     (I)nvestment
     (A)PR
     (L)IFE
Enter your choice ?
```

You must select the unknown to be found. There are four options. Listed below are four similar runs (questions and answers) solving for the unknowns in the order listed above:

Enter **P** <CR> yields a transfer to the CALL BALLOON NOTE VALUE subroutine and will display the following on the screen:

```
***** CALL BALLOON NOTE VALUE PROGRAM *****
PRESENT VALUE-INVESTMENT ? 1000 <CR>
APR ? 12 <CR>
LIFE IN YEARS ? 4 <CR>
PAYMENTS PER YEAR ? 12 <CR>

OUTPUT INFORMATION
PAYMENT:          26.33
LOAN:         1,000.00
LIFE:    4
PAYMENTS/YEAR: 12
12%
<CR> TO CONTINUE ? <CR>
```

The <CR> will return you to the main EVALPAC menu. If you desire to return to the MORTGAGE program, enter **M** <CR> and the MORTGAGE main menu will appear again. This time select the "**I**" option.

You will see:

```
INPUT DATA
PAYMENT ? 26.33 <CR>
```

```
APR ? 12 <CR>
LIFE IN YEARS ? 4 <CR>
PAYMENTS PER YEAR ? 12 <CR>
```

---

```
OUTPUT INFORMATION
PAYMENT:          26.33
LOAN:           999.85
LIFE:  4
PAYMENTS/YEAR:  12
12%
<CR> TO CONTINUE ? <CR>
```

To test the APR calculation, return to the MORTGAGE program and select the **A** option. The prompts are:

```
INPUT DATA
PAYMENT ? 26.33 <CR>
PRESENT VALUE - INVESTMENT ? 1000 <CR>
LIFE IN YEAR ? 4 <CR>
PAYMENTS PER YEAR ? 12 <CR>
LINEAR APPROXIMATION USED FOR APR
<CR> TO CONTINUE ? <CR>
```

---

```
OUTPUT INFORMATION
PAYMENT:       26.33
LOAN:      1,000.00
LIFE: 4
PAYMENTS/YEAR: 12
 11.98758267340959 %
<CR> TO CONTINUE ? <CR>
```

The calculation for an unknown life is as follows:

```
INPUT DATA
PAYMENT ? 26.33 <CR>
PRESENT VALUE - INVESTMENT ? 1000 <CR>
APR ? 12 <CR>
PAYMENTS PER YEAR ? 12 <CR>
```

---

```
OUTPUT INFORMATION
PAYMENT:         26.33
LOAN:        1,000.00
LIFE:  4
PAYMENTS/YEAR:  12
 12%
<CR> TO CONTINUE ? <CR>
```

In general, the item being calculated is a little off. The payment is assumed to be correctly calculated as $26.33. This may or may not be the case. It is more likely that the actual payment that would make everything come out even is some value between 26.334999999999999... and 26.325, instead of 26.33. Such an "error" is enough to account for a present value of $999.85 rather than $1,000.00 or an interest rate of 11.98758267340959 percent rather than twelve percent. In the case of years, the value came out exactly right.

## 17.2 Theory

The equations used to calculate the payment, investment, and life are found in most finance books. There is no equation that will yield the annual percentage rate. To find this value, you may use a direct search trial and error method or some indirect search method. Since the value sought is not an extreme value, the normal search routines used to find optimals cannot be used.

The method used to find the annual percentage rate is a linear approximation technique. Two points are selected, one near zero the other near one. A linear line is projected between the two points. The point where this line crosses the axes is determined. Next, the program moves away from this point and selects two more points. A linear line again is projected between these two points, and the point where it crosses the axes is determined. Again, the program moves away from this point but for a shorter distance than the first time. This process is repeated five times. This series of calculations results in a value very close to the value input for the annual percentage rate. Take care that the accuracy of this approximation is reduced as the value of the interest per period approaches zero or one.

## 17.3 Examining The Program

The computer is first routed to line 410, where the user is asked to select the unknown being solved for in lines 420 through 440. Lines 450 through 480 are used to find the value of SM$ to be used in the input statements in lines 490 through 530.

Line 540 routes the computer to line 560 for the call routine. This line will not be used in the mortgage routine.

Lines 550 through 580 are used to calculate the payment, if that is the unknown. Lines 590 through 610 solve for the present value. Lines 620 through 680 are the lines that do the calculations to find the annual percentage rate. Lines 690 through 700 solve for the life of the loan. Lines 710 through 770 print the results on the monitor. When the printing is complete, the program is returned to the main menu by line 770.

# 18. (P)resent Value Of Unequal Cash Flows

In most business ventures, annual cash flows are seldom equal due to the nature of the depreciation methods and financing procedures normally used. In all depreciation methods except the straight line method, the annual amount depreciated is different from year to year. In all financing methods, with the exception of the interest only method, the amount of dollars paid with taxed dollars rather than pretaxed dollars is different from year to year. This means the taxable income, and therefore the amount of tax paid, is a variable from year to year.

## 18.1 Running The Program

After you have selected the "P" option, the first thing you will see is:

```
DISCOUNT RATE (%) ?
```

The methods used to select the discount rate, internal rate of return, and yield are beyond the scope of this book. Financing books spend many pages on how best to perform this task. Ten percent was selected as the value to be used in the example. Type **10** <CR> and the next thing you will see is:

```
(C)RT or (P)rinter ?
```

The printer option was selected, because the results are a bit different from the CRT option and we wish to illustrate this fact. The next question is:

```
YEAR     CASH FLOW AMOUNT ($)
```

We have selected a tabular data entry format for convenient data entry. To terminate the data entry simply enter a **0** under the year column and a **0** for the cash flow amount. This will terminate the data entry sequence.

Assume we have a three year unequal cash flow comprised of the following cash flow amounts:

| Year | Cash Flow Amt. |
|---|---|
| 1 | 1000.00 |
| 2 | 2000.00 |
| 3 | 3000.00 |

Given the above data, we will examine the microcomputer data entry for this problem.

After selecting the printer as the output option, the screen will display the following:

```
YEAR      CASH FLOW AMTS ($)
1 <CR>    1000 <CR>
2 <CR>    2000 <CR>
3 <CR>    3000 <CR>
0 <CR>    0 <CR>
```

By programming the input in this manner, the program allows you a great deal of control over what the program does. Listed below is a copy of the CRT input and output followed by the output obtained on the printer.

```
CRT Output
PRESENT VALUE : $4815.92765015787
<CR> TO CONTINUE ? <CR>
```

The output to the line printer is:

```
DISCOUNT RATE 9.999999776482582
YEAR                VALUE               SUM OF DISCOUNTED VALUES
 1                   1000               909.0908893868947
 2                   2000               2561.983399260576
 3                   3000               4815.92765015787
```

All output was performed without USING statements. The numbers were allowed to take on whatever values they were assigned in the program. This was done to allow you to determine how and when the values should be rounded, since there was no way to predict the many possible applications to which this routine might be utilized.

## 18.2 Theory

This program is based on the time value of money idea that the future value is equal to the present value plus the interest earned by the present value. Turning this idea around, the present value may be calculated by dividing the future value by one plus the interest per period stated as a decimal, the quantity raised to the number of periods it is being discounted.

Specifically,

```
present value = future value / ( 1 + interest) periods
```

This relationship is reviewed in depth in most finance or engineering economy text books.

## 18.3 Examining The Program

Line 320 sends the computer to line 1520 when this option is selected. Line 1520 is a REMark statement that identifies the beginning of the routine.

In lines 1530 and 1540, the discount rate is input and the selection of CRT versus printer is made. The variable SUM# is set equal to zero.

In line 1540, the headings are printed on the printer if the hard copy option was selected. Line 1560 is the question containing the year and the amount and is repeated over and over again. The program calculates in lines 1570 through 1600.

When you input 0,0, line 1570 directs the program to line 1610 to print the final results on the monitor. The results appear on the printer as they are generated.

Line 1630 returns you to the main menu when the routine is complete.

# 19. (R)eal Estate Property— 1981 Depreciation Law

The table lookup method developed for depreciation in the Tax Reform Act of 1981 is one of the most efficient methods of performing these calculations when they are performed manually. In the development of the tables, the rules could not be determined and to do this activity on the computer, it was necessary to read the Internal Revenue Service percentage depreciation rates into the computer in the form of DATA statements.

The system presented herein is officially called the 1981 Accelerated Cost Recovery System or ACRS for short. In 1982, some significant modifications were made to the personal property section of ACRS but left intact most of the real property provisions that were enacted in August of 1981.

Presently, there are two sections of the real property tax code. The first is the "Low Income Real Property" and the second is called "Other Real Property." Most business applications deal with the "Other Real Property" category. Our program considers both classes of real property.

## 19.1 Running The Program

The first thing you will see on the monitor after selecting the "R" option is:

```
***** 1981 REAL PROPERTY DEPRECIATION *****
(C)RT, (P)rinter ?
```

In the 1981 law, separate tables are provided for real property and personal or production property. This routine identifies itself as for real estate property use only.

The standard life of a real estate asset is fifteen years. It is possible to print the entire depreciation table on your CRT without using a printer option. The printer option was included to allow you to input identification information and in case the tax laws change again.

Select "C" for the CRT option. The next question is:

```
FIRST COST ?
```

First cost and capital cost are the same thing. (Technically, the Federal Tax Code uses the term "basis.") This is the amount of dollars that cannot be expensed and must be capitalized according to the current law. Select $1,000 as the value of the asset to be depreciated. Type 1000 <CR> and you will see:

```
YEAR (19XX) ?
```

Copies of the tables included in this program are found in the theory section. To answer the above question, you must type in the entire year. A simple 88 will not produce results. 1988 must be typed for that year. Type 1988 <CR> and you will see:

```
LIFE OF ASSET IS 15 YEARS ONLY
MONTH PURCHASED 1,2,... 12 ?
```

The life of the asset is assumed to be fifteen years. The amount of depreciation for each month will vary. This is a fixed month-by-month allocation method. There is no advantage in waiting until the end of the year to purchase an asset. You can only depreciate the asset based on the month in which the asset was purchased.

Use 12 as your answer to the above question. The next question is:

```
LOW INCOME HOUSING (Y/N) ?
```

The government has long been using depreciation as a method of getting individuals to invest in selected activities. The 1981 law is no exception. Although not great, there is an advantage to investing in low income housing. This is evidenced by the slightly higher ACRS recovery percentages for low income housing as opposed to the same ACRS percentage rates for other real property. For the example answer Y <CR>.

The computer will now produce:

| YEAR | DEPRECIATION | BOOK VALUE |
|---|---|---|
| 1988 | 10.00 | 990.00 |
| 1989 | 120.00 | 870.00 |
| 1990 | 100.00 | 770.00 |
| 1991 | 90.00 | 680.00 |
| 1992 | 80.00 | 600.00 |
| 1993 | 70.00 | 530.00 |
| 1994 | 60.00 | 470.00 |
| 1995 | 60.00 | 410.00 |
| 1996 | 60.00 | 350.00 |
| 1997 | 50.00 | 300.00 |
| 1998 | 50.00 | 250.00 |
| 1999 | 50.00 | 200.00 |
| 2000 | 50.00 | 150.00 |
| 2001 | 50.00 | 100.00 |
| 2002 | 50.00 | 50.00 |
| 2003 | 50.00 | 0.00 |

```
<CR> TO CONTINUE ?
```

## 19.2 Theory

The theory associated with the use of this depreciation method is very simple. The Internal Revenue Service has published a set of tables. You must use this set of tables to determine the depreciation you are able to charge in any given year. The values printed within the tables represent annual percentages that are used by multiplying the original recovery basis by the percentage tabulated in the the table.

The tables for "Other Real Property" and "Low Income Housing" are represented in Tables 19.1 and 19.2.

| Recovery | Month Unit Placed in Service (Regular) | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Year | Jan | Feb | Mar | Apr | May | Jun | Jul | Aug | Sep | Oct | Nov | Dec |
| 1 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| 2 | 10 | 10 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 12 |
| 3 | 9 | 9 | 9 | 9 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 |
| 4 | 8 | 8 | 8 | 8 | 8 | 8 | 9 | 9 | 9 | 9 | 9 | 9 |
| 5 | 7 | 7 | 7 | 7 | 7 | 7 | 8 | 8 | 8 | 8 | 8 | 8 |
| 6 | 6 | 6 | 6 | 6 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 |
| 7 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 |
| 8 | 6 | 6 | 6 | 6 | 6 | 6 | 5 | 6 | 6 | 6 | 6 | 6 |
| 9 | 6 | 6 | 6 | 6 | 5 | 5 | 5 | 5 | 5 | 6 | 6 | 6 |
| 10 | 5 | 6 | 5 | 6 | 5 | 5 | 5 | 5 | 5 | 5 | 6 | 5 |
| 11 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| 12 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| 13 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| 14 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| 15 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| 16 | 0 | 0 | 1 | 1 | 2 | 2 | 3 | 3 | 4 | 4 | 4 | 5 |

Table 19.1: Other Real Property Percentage Recovery Rates

| Recovery | Month Unit Placed in Service (Low Income Housing) | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Year | Jan | Feb | Mar | Apr | May | Jun | Jul | Aug | Sep | Oct | Nov | Dec |
| 1 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 4 | 3 | 2 | 1 |
| 2 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 13 | 13 | 13 | 13 | 13 |
| 3 | 10 | 10 | 10 | 10 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 |
| 4 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 10 | 10 | 10 | 10 |
| 5 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 9 |
| 6 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 |
| 7 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 |
| 8 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 6 | 6 |
| 9 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| 10 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| 11 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| 12 | 4 | 4 | 4 | 5 | 4 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| 13 | 4 | 4 | 4 | 4 | 4 | 4 | 5 | 4 | 5 | 5 | 5 | 5 |
| 14 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 5 | 4 | 4 |
| 15 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 16 | 0 | 0 | 1 | 1 | 2 | 2 | 2 | 3 | 3 | 3 | 4 | 5 |

Table 19.2: Low Income Housing Recovery Percentages

## 19.3 Examining The Program

The "R" selection is executed in line 360 of the menu section. The program is sent from this position to line 2750. Line 2750 is a REMark statement that identifies the beginning of the routine.

Line 2750 prints the heading for the program and sends the computer to line 2860, where the selection of the type of output is made. First cost is input on line 2870 followed by the year in line 2880. Also, in this line, the RESTORE statement puts the program at the beginning of the DATA statements, and then the 99 data values needed for personal property depreciation are read.

In line 2900, the fifteen year life specification is printed. In line 2910, the month in which the asset was purchased is read into the program.

Line 2920 is used to make sure the number of the month specified is within the limits 1 to 12. Line 2930 reads in the type of housing, and line 2940 begins executing the calculations needed to produce the depreciation schedule. Notice that most of the work in this program was to make sure the correct data was read. Lines 2980 through 3220 are the data corresponding to the tables reviewed in the theory section.

# 20. (S)traight Line Depreciation

Straight line depreciation represents the simplest form of capital recovery. All other approaches are compared to straight line. To find the depreciation per period, the first cost (basis) is identified and the salvage value is estimated. The difference between these two values is calculated and divided by the number of years involved in the recovery. Since the basis and the salvage value are explicit in the calculations, the straight line method of depreciation will recover the asset down to the estimated salvage value.

## 20.1 Running The Program

After calling in the straight line depreciation routine from the main menu you will see:

```
***** Straight Line Depreciation *****
(C)rt or (P)rinter ?
```

For short runs, either the screen copy technique or the hard copy routine may be used to get printer copy. Select "**C**" for the sample run.

The next question is:

```
YEAR OF PURCHASE ?
```

The straight line technique has been an acceptable method of depreciation for many years. Even under the Tax Reform Act of 1981, it is maintained as an acceptable alternate method of calculation. Input **1984** <CR> as the year in answer to the above question.

```
DEPRECIABLE BASIS ? 1000 <CR>
```

As shown, the value of $1,000 was assigned to the recovery basis for the sample problem. The next question is:

```
SALVAGE VALUE ? 100 <CR>
```

It is possible to depreciate many assets to zero salvage value under the current tax law. For this example, a salvage value of $100 was used.

```
DEPRECIABLE LIFE ?
```

A five year life was selected for the sample problem. Type **5** and push the <CR> key.

The next question is:

```
MONTH OF PURCHASE ?
```

Use **6** months to examine the allocation method built into the program. After you have pressed <CR>, the computer will produce the following results:

| YEAR | DEPRECIATION | CUMULATIVE | BOOK VALUE |
|---|---|---|---|
| 1984 | 180.00 | 180.00 | 820.00 |

```
1985            180.00          360.00          640.00
1986            180.00          540.00          460.00
1987            180.00          720.00          280.00
1988            180.00          900.00          100.00

DO YOU WISH MONTHLY ALLOCATION (Y OR N)? Y <CR>
```

The above question was answered "y." As a result, the following will be produced:

```
YEAR          DEPRECIATION          CUMULATIVE          BOOK VALUE
1984                105.00              105.00              895.00
1985                180.00              285.00              715.00
1986                180.00              465.00              535.00
1987                180.00              645.00              355.00
1988                180.00              825.00              175.00
1989                 75.00              900.00              100.00
```

Be careful to check with your accountant on what the current regulations are relative to the allocation of depreciation amounts before you use the program.

## 20.2 Theory

Straight line depreciation assumes a constant decline in the book value of a capital asset. As noted in the introduction, the calculation method consists of finding the difference between the first cost and the salvage value and then dividing by the number of periods.

## 20.3 Examining The Program

This program uses much of the same code used by the declining balance depreciation and sum of the years digit depreciation methods. After you select "S" in the main menu, you will be transferred to line 1640. In line 1650, you are asked if you need printer output or not.

The string variables used to define the different input data are defined in lines 1660 through 1730. Lines 1740 through 1770 are used to input the data needed to calculate the depreciation expense.

Lines 1780 through 1820 initialize some of the variables needed in the calculations. Line 1830 is used to input asset identification information if a printer is used to produce the output. Line 1840 starts the printer output of the asset identification information as needed.

Lines 1850 through 1900 define some additional initial values of variables and print some more information on the printer. Line 1910 routes the straight line program to line 2170.

Lines 2170 through 2190 do all the straight line depreciation calculations, while the monthly allocation is done in lines 220 through 2290. In all cases, lines 2310 through 2350 are a subroutine used to print the information on both monitor and printer.

# 21. (V)alue Discounted of a Bond or Mortgage

Bonds and mortgages are sold regularly. Their prices depend on the cash flow of the item, the present market yield, and interest rate. This program handles both bonds and mortgages. Mortgages with and without balloon notes may be handled.

## 21.1 Running The Program

The program starts with the selection of the type of loan to be sold:

```
***** DISCOUNTED BOND OR MORTGAGE ANALYSIS *****
(B)OND OR (M)ORTGAGE ?
```

The cash flow on a bond and that on a mortgage are completely different. In the theory section (21.1), we will review the assumptions underlying this program. For this first run, select "**B**" for bond. Type **B** <CR>, and you will see:

```
FACE VALUE OF BOND ?
```

Assume the original face value of the bond was $1,000. This means that when the bond matures, you will get the $1,000 back.

The next question is:

```
STATED RATE OF RETURN ?
```

The stated rate of return on a bond determines how much the bond pays each period. It has no other meaning of interest in this type of analysis. Assuming the stated rate of return was twelve percent, type **12** <CR> and the next question will appear:

```
LIFE OF BOND ?
```

Assume, for this example, a life of five years. Type **5** <CR> to continue:

```
PAYMENTS PER YEAR ?
```

Bond interest may pay annually, biannually, quarterly, or monthly. In this case, the bond pays twice a year. Input **2**<CR>.

```
MARKET RATE OF RETURN ?
```

The price of a loan instrument depends on the cash flow and the market rate of return. The market rate of return takes into account factors such as whether the loan is for short-term or long-term, the safety of the investment, and so on. In this case, the market rate of return for this security is 18 percent. Type **18** <CR> to enter this data.

The program now computes the following:

```
CURRENT VALUE:
807.47
<CR> TO CONTINUE ?
```

Since the rate of return to be earned was greater than the rate earned on the loan, the price must be less than the original price. The results are as expected.

If you had chosen to evaluate a mortgage you would have the option of considering a mortgage with or without a balloon payment. Listed below are two runs of this routine. Each of the two options was selected.

With a balloon payment:

```
(B)OND OR (M)ORTGAGE ? M <CR>
PRESENT VALUE - INVESTMENT ? 1000 <CR>
APR ? 12 <CR>
LIFE IN YEARS ? 30 <CR>
PAYMENTS PER YEAR ? 12 <CR>
MARKET RATE OF RETURN ? 20 <CR>
BALLOON NOTE (Y/N) ? Y <CR>
```

After pressing <CR>, the program logic will produce the following results on the screen:

```
PAYMENT # OF BALLOON PAYMENT ? 120 <CR>
PAYMENT: 10.29
LOAN:          1,000.00
LIFE: 30
PAYMENTS/YEAR:  12
 12%
BALLOON NOTE              934.18
BALLOON PAYMENT # 120
CURRENT VALUE:
            615.56
<CR> TO CONTINUE
```

Without a balloon payment:

```
(B)OND OR (M)ORTGAGE ? M <CR>
PRESENT VALUE - INVESTMENT ? 50000 <CR>
APR ? 6 <CR>
LIFE IN YEARS ? 30 <CR>
PAYMENTS PER YEAR ? 12 <CR>
MARKET RATE OF RETURN ? 18 <CR>
BALLOON NOTE (Y/N) ? N <CR>
CURRENT VALUE:
          19,891.05
<CR> TO CONTINUE
```

In the case of the mortgage with a balloon payment, there is a lot of information printed in addition to the current value of the loan. This information is generated by the program that handles balloon payments. To calculate the current market value of a loan of this type, the payments and balloon payment must be known. Notice the $1,000 mortgage has monthly payments of $10.29, a balloon payment of $934.18, and a current market value of $615.56.

The mortgage investment without a balloon note for a 30-year term with a face value of $50,000 and a yield of 6 percent, has a current value of $19,891.05, if the market rate of return is 18 percent.

## 21.2 Theory

To find the present value of a loan, you must first identify the cash flows associated with the loan. For a bond, the cash flows consist of the period payments and the payment of the face value of the bond when it becomes due and payable.

In a mortgage, the cash flow consists of the periodic payments only. Usually these payments are made monthly. In the case of a balloon payment, mortgage cash flows consist of the periodic payments plus the balloon payments.

It is possible to identify the cash flows and use other routines in the EVALPAC set of programs to find the present value of the cash flows. The advantage of this routine to calculate the present value of loans is that the work is performed automatically.

## 21.3 Examining The Program

Line 370 routes the computer to line 3230 when the present value of a loan option is selected. The first line in this routine is a REMark to identify its starting point. Line 3240 clears the screen, and the choice of bond or mortgage is made in line 3250.

If the mortgage option is selected, line 3260 routes the computer to line 500, where the mortgage payments are calculated. In the case of a bond, the face value of the bond is read in line 3270, while the stated rate of return is read in line 3280.

Lines 3290 through 3330 input the life of the bond and the market rate of return and perform all the calculations needed for the bond analysis. The program then is routed to line 3370 for the bond case.

After the mortgage information is input and the values are calculated in lines 500 through 570, the program returns to line 3340, where the market rate of return is input and the calculations of present value start.

Line 3350 contains the question relative to the balloon note. If the answer is yes, the program is routed to line 1390 where the balloon note calculations are performed. When this work is done, the program goes to line 710 where the balloon information is printed. Line 750 returns the program to line 3360, where the calculations are completed.

The current value is printed in line 3370 for all alternatives, and the routine is returned to the main menu in line 3390.

# 22. (D)epreciation Personal Property 1981 Law With 1982 Revisions

Under the Economic Recovery Tax Act Of 1981, four classes of personal property were assigned three, five, ten, and fifteen year tax lives. Three depreciation schedules for 1981 through 1984, 1985, and 1986 or beyond were published. In 1982 the law was revised so that the 1981 tax schedule would be used without change. This program produces depreciation schedules based on the tables for each class of property. The 1981 schedule is used, although the old schedules are still in the program but have not been implemented in the program logic.

This program is used as an aid in the calculation of capital recovery amounts for personal property placed in service on or after January 1, 1981. To use the program you must know the class of property you are dealing with. Four specific classes exist: three-year property, five-year property, ten-year property and fifteen-year property. Some examples of items in each class are:

Three-Year: Automobiles, pickup trucks, certain research and development items, and some "special purpose" tools.

Five-Year: General purpose tools and most productive machine tools.

Ten-Year: Some special purpose buildings.

Fifteen-Year: Most real property buildings and structures.

Special Note: According to the 1982 Tax Law, if one elects to claim the investment tax credit on qualified property (specifically three-year property and most five and ten-year property excluding real property) then the following rules apply:

Ten percent Investment Tax Credit-applies to five and ten year property. However, if claimed, the basis of the property must be revised downward by 50 percent of the ITC amount.

Example: An item costs $1000.00 and is eligible for the 10 percent ITC. The tax credit will be $100.00 ($1000.00 x 0.10). The basis for capital recovery will be:

$$\$1000.00 - (0.10 \times \$1000.00)/2 = \$950.00.$$

For capital recovery purposes, the $950.00 would be used to compute the depreciation amounts...not the $1000.00 original purchase price. The $950.00 is called the "revised basis."

Eight percent Investment Tax Credit-applies to five- and ten-year property but the original basis is not revised downward. Thus, the taxpayer can elect either the full ten percent ITC rate with basis adjustment or the lower eight percent rate without any basis adjustment. For three-year recovery property, the ITC is six percent with basis adjustment or the alternate four percent ITC rate without basis adjustment.

The program considers only the basis amount and, as written, does not compute the Investment Tax credit.

## 22.1 Running The Program

The program starts by identifying itself and asking what type of output is desired:

```
1981 DEPRECIATION RULES-PERSONAL PROPERTY WITH 1982 REVISIONS
(C)RT (P)rinter ?
```

Select C <CR> for output to the screen of your monitor. The next question is:

```
DEPRECIABLE BASIS = ?
```

Use $1,000 for the first cost of the sample asset. Type 1000 <CR> to continue to the next question:

```
YEAR (19XX) ?
```

In this program, as in the case of real property under the 1981 law, the entire year must be entered. Type 1988 <CR> as your response for the sample problem.

The next question is:

```
LIFE OF ASSET
  3   5   10   15
?
```

The program is written so that you must choose 3, 5, 10, or 15. These are the only acceptable lives for personal assets under the current law. Type 10 <CR> for this sample problem.

The computer now will produce the following results:

| YEAR | DEPRECIATION | BOOK VALUE |
|---|---|---|
| 1988 | 80.00 | 920.00 |
| 1989 | 140.00 | 780.00 |
| 1990 | 120.00 | 660.00 |
| 1991 | 100.00 | 560.00 |
| 1992 | 100.00 | 460.00 |
| 1993 | 100.00 | 360.00 |
| 1994 | 90.00 | 270.00 |
| 1995 | 90.00 | 180.00 |
| 1996 | 90.00 | 90.00 |
| 1997 | 90.00 | 0.00 |

<CR> TO CONTINUE?

## 22.2 Theory

The theory of this approach to depreciation is based completely on the tables published by the government. The original 1981 tables that have been built into the program are shown below:

For property placed in service after December 31,1980, the following percentage rates are currently in force:

| Year | 3-year | 5-year | 10-year | 15-year |
|---|---|---|---|---|
| 1 | 25 | 15 | 8 | 5 |
| 2 | 38 | 22 | 14 | 10 |
| 3 | 37 | 21 | 12 | 9 |
| 4 | | 21 | 10 | 8 |
| 5 | | 21 | 10 | 7 |
| 7 | | | 9 | 6 |
| 8 | | | 9 | 6 |
| 9 | | | 6 | 6 |
| 10 | | | 9 | 5 |
| 11 | | | | 6 |
| 12 | | | | 6 |
| 13 | | | | 6 |
| 14 | | | | 6 |
| 15 | | | | 6 |

**Table 22.1: 1981 ACRS Standard Percentage Recovery Rates for Personal Property**

The schedule provides for a fixed amount of depreciation, no matter when in the year the asset is purchased. This means that if you buy a piece of equipment on January 1 or December 15 you will be allowed the same tax writeoff for the year.

## 22.3 Examining The Program

Line 350 transfers the flow of the program to line 2360 when an X is given as the answer in the main menu. Line 2360 is a remark statement included to mark the beginning of the routine. In lines 2370 and 2440, the identification information is printed, and the question relative to what type of output is asked.

Lines 2450 through 2530 input the remaining information needed and perform the task of checking if acceptable information has been input. In line 2540, the program is passed to a subroutine that prints headings and such, and all DATA statements are set at their beginning with a RESTORE statement. Also, in this statement some data is read depending on which year is of interest.

In line 2550, 2580, 2600, or 2620, the actual data to be used is read, and in line 2560 the data is again restored to the beginning. Line 2560, using a subroutine for printing, does all the calculations needed in the program.

The data for this program is found in lines 2630 through 2740. This is the same data as found in the tables reproduced above. The reason this data was not removed is that it would have thrown the data count off for all routines that use DATA statements.

# 23. (+)Sum Of The Year's Digits Depreciation

This method of depreciation is selected in the main menu by typing + <CR>. The reason the plus sign was used is that "S" was used for straight line depreciation.

This depreciation technique is an accelerated method. It is no longer used by the Federal government under the Tax Reform Act of 1981 but is still required by some states.

## 23.1 Running The Program

After typing + <CR>, the next thing you will see on your monitor is:

```
Sum Of The Year's Digits Depreciation
(C)RT, (P)rinter ?
```

The same procedure and the same code is used for all three of the traditional methods of depreciation (straight line, declining balance, and sum of the year's digits). If you have a long run or wish to use the identification information generated for the printer output, "P" should be selected. Use "C" for screen output.

The next question you will see is:

```
YEAR OF EXPENSE ?
```

The program allows you to identify each line of output relative to the year to which it is associated. This is very useful as the asset ages in identifying where on the depreciation schedule you are at any given time. Use **1987** for the example.

The next question is:

```
DEPRECIABLE BASIS = ?
```

The value of **$1,000** has been used for all our depreciation examples. Continue with this practice. Remember that what is included in capital expense is a matter of law, and the most current law should be used when determining this item at any point in time.

The program continues with the question:

```
SALVAGE VALUE ?
```

It is possible to depreciate many items to zero salvage value under the Tax Reform Act of 1981. In a real estate investment, it is still necessary to separate land value from that of the improvements. Land may not be depreciated. Use **$100** as the salvage value for the sample run.

```
LIFE ?
```

The life selected for the sample run was five. Long runs were not selected to keep the amount of output within the limits of the monitor and to keep the output within reason.

The last question is for monthly allocation of depreciation:

```
MONTH OF PURCHASE ?
```

The program is designed to allocate to the nearest month. If you are using the six months convention, be sure to input the month of purchase in such a manner as to reflect the period of interest. Use **seven** for the example.

The computer will now produce the following results:

```
YEAR          DEPRECIATION          CUMULATIVE          BOOK VALUE
1987                300.00              300.00              700.00
1988                240.00              540.00              460.00
1989                180.00              720.00              280.00
1990                120.00              840.00              160.00
1991                 60.00              900.00              100.00
DO YOU WISH MONTHLY ALLOCATION (Y OR N)?
```

The ability to allocate monthly is one of the nicer features of this program. This allocation is performed manually in many depreciation programs.

If you answered "Y" to the above, you would obtain the following results:

```
YEAR          DEPRECIATION          CUMULATIVE          BOOK VALUE
1987                150.00              150.00              850.00
1988                270.00              420.00              580.00
1989                210.00              630.00              370.00
1990                150.00              780.00              122000
1991                 90.00              870.00              130.00
1992                 30.00              900.00              100.00

<CR> TO CONTINUE ?
```

The allocation procedure adds an additional year to the depreciation schedule. The values in this year depend on when during the year of purchase the asset was purchased.

## 23.2 Theory

If you add the numbers from 1 through 12, you will get 76. The sum of the year's digits is referred to as the rule of 78 for this reason, when it is used for allocation within a given year. If you have an asset with a five-year life, the sum of the year's digits is 1+2+3+4+5 or 15. The depreciation is allocated 5/15 the first year, 4/15 the second year, etc. The method is simple but time consuming—a perfect job for the computer.

## 23.3 Examining The Program

Line 340 selects the "+" as an acceptable answer in the main menu. The title is defined in this line, and then the flow of the program is transferred to line 1640. Lines 1640 through 1760 are shared with the other traditional methods of depreciation. In these lines, the string variables used for input and output

are defined, and in lines 1740 through 1760 all the input information is read into the computer. Line 1770 is used for the declining balance method of depreciation only to read in the rate of the depreciation.

Lines 1780 through 1800 continue to initialize values, while line 1810 defines the printer output variable D(8) as one, if printer output has been called for.

Line 1820 continues to define string variables for output, while line 1830 starts input of identification type information, if printer output has been specified.

Line 1840 outputs printer identification information. Lines 1850 through 1880 continue to initialize both numeric and string variables. Line 1890 starts to print headings for screen output, while line 1900 prints the same thing on the printer if it is asked for.

Line 1910 is used to route the straight line depreciation routine to its code, while line 1920 routes the sum of the digits method to line 2070.

The sum of the year's digits calculations are performed in lines 2070 through 2160. Notice that in line 2150, the printing subroutine in lines 2310 through 2350 is called in to print the results on both printer and screen.

When the program has completed the printing of the sum of the year's digits depreciation, it proceeds to line 2200 to ask the question about monthly allocation. If the answer is yes, then the code in lines 2220 through 2300 is used to calculate the monthly allocation, and the subroutine in lines 2310 through 2350 is called in to print the results.

# 24. (C)ash Flow Analysis

The collection of programs in EVALPAC is written to allow an individual to make almost any kind of business/financial analysis. Since control of the procedures and method rests with the individual, this approach is of great value to the knowledgeable investor who wants help with the many calculations required in a business investment.

In this chapter, we have developed a program that will perform the bulk of the calculations needed to assist in the analysis of a business investment. The program requires the user to input data, interpret results, and then apply the decision indicated by the analysis.

The format of the program is aimed towards real estate investments. It can be used for other investments, but care must be taken in the input of data, the use of depreciation schedules, and the evaluation of the results obtained.

The program is essentially for after-tax cash flow evaluation. It first identifies the cash flows and then finds the net present value of all flows based on the Net Present Value concept.

The user need not be knowledgeable in the theory or calculations. He needs only to input data and understand the results.

The model is as accurate as we could make it, but it is still a model. All models have approximations and assumptions. This one is no exception. Establish a procedure for using the model, then examine the results using the model on completed projects.

Compare the decisions you made with the decision the program recommends. If necessary, adjust your procedure for using the model until it becomes a useful predictor, then use it on a trial basis.

Computers can turn out wrong answers with amazing speed. Be very careful when entering the data for the program. If you input questionable data into the program you will get misleading results and could lose a great deal of money. Always keep in mind the risk that relates to the value of the data being entered into the program.

One investor has been using this model for about ten months. He has saved money by rejecting poor projects, although errors do occur. One out of thirteen business deals failed to behave as predicted. An after-the-fact examination of the investment project indicated that the poor results came from inputting faulty information into the model. As a matter of fact, when the actual sales figures were keyed in, the model predicted the results obtained very closely.

A track record of twelve out of thirteen, 92.3 + percent, is good for any decision making tool. Compare this track record with yours. Do you have a use for the program? Even if your track record is better than ours, you may wish to have a computer take over the task of making the many calculations needed in business decision making.

Investment analysis is not a simple activity. You cannot expect a simple answer to a complex problem. If you need a first approximation to an invest-

ment situation, the break-even program that is part of the EVALPAC set may be the best approach for you. If you need an accurate and detailed approach to decision making, this program may be better.

## 24.1 Sample Application

The sample application is based on a real estate investment involving a 44-unit apartment complex. It has the potential ability to earn $96,000 per year. The vacancy rate is estimated to be approximately five percent. To purchase this complex $100,000 cash is needed. A 25-year mortgage for $600,000 to cover the remaining cost will be issued by the seller at ten percent. The estimated selling price of the investment in ten years is $700,000.

When the complex is taken over, initial repair costs are estimated to be $15,000. Annual maintenance costs are estimated to be $25,000. Based on the Economic Recovery Act of 1981, a table lookup method of depreciation may be used for this type of investment. This method is built into the program along with other methods of depreciation. A land value of ten percent of the investment is estimated.

Selecting the study period is very important. It does make a difference. A ten-year study period was chosen, because it balances the effect of the annual cash flows with the effect of the resale estimate at the end of the study period. In practice, the selection of a "standard" study period for specific types of investments will help make the program a constant measurement tool to judge investments of the specific type.

An internal rate of return or yield of fifteen percent will be used for this analysis. The number has nothing to do with the cost of capital above. It is a number that represents the expectation of local investors in the specific type of investment under consideration. Again, the use of a "standard" or consistent value is most important in using the program as a measurement tool.

Inflation is an emotional subject to many investors. This program forces you to consider inflation as just one more item to be forecast. It will show you the consequences of different inflation estimates. At least two evaluations, one at zero inflation and one at six percent or some other expected inflation rate, is recommended. Such a procedure will be followed in this example.

For this example, a tax bracket of 50 percent for the investor has been assumed. The tax shelter aspects of a real estate investment are very important and the higher an individual's tax rate, the more shelter will be obtained from a given investment. It is not unusual to find that an investment that is good for someone in the 50 percent tax bracket is no good for someone in the 20 percent bracket.

## 24.2 Running The Program

The program is called from the main menu program by typing **C**<CR>. The first thing you will see is:

```
***** APPLE CP/M CASH FLOW ANALYSIS PROGRAM ***
                    by
          Steven M. Zimmerman Ph.D.
                 Leo Conrad
          Donald R. Smith Ph.D., P.E.
            Press < CR > to Begin
```

After pressing <CR> the second screen will appear as:

```
CASH FLOW ANALYSIS PROGRAM FOR APPLE CP/M
ENTER ALL PERCENTS AS WHOLE NUMBERS--NOT AS DECIMALS
TITLE : ?
```

The note about inputting percents as whole numbers rather than decimals is very important. It is easy to make an error and input the wrong data for this type of number unless a standard procedure is developed and followed.

It is assumed that your output will be printed on a printer. It is important to have both the questions and answers available on a printed copy of computer output. A week after a run is made it is very often difficult to identify a printout unless a title is included.

Use the title "**44 Unit Apartment 123 West Drive**." Type the title followed by a <CR>, and you will see the next question:

```
DATE :(XX/XX/XX) ?
```

The same property may be considered for investment more than a single time. Knowing the date of the last analysis helps in determining if enough changes have occurred to justify a change in the decision to buy the investment. It is also useful if you are called six months or a year later about an old offer, to see if you are willing to renew the offer. Type **08/19/83** <CR> to continue. The next question relates to the study period.

```
STUDY PERIOD IN YEARS ?
```

It is strongly recommended that you select one study period and stay with it. There is one exception to this rule. That is in the case of land investment deals where the only significant income occurs when the investment is sold. In this case the intent of the evaluation should be to identify the number of years the investment can be held and still yield the stated rate of return with some given target price. In other words you are seeking an investment window. In this case, the maximum years should be selected as some large number beyond where you think the window will exist. In the case of the 44-unit apartment investment the ten-year standard time frame was selected. Type **10** <CR>.

The screen will clear, and the first menu will appear on your monitor:

```
                *** INCOME MENU ***
            INPUT MENU REGULAR INCOME
(C)onstant
(R)edo all regular income
(S)pecified year-by-year
(E)ND CYCLE?
Enter Your Choice (C/R/S/E) ?
```

(C)onstant income is income that may be estimated as a single number for the length of the ten-year investment span. If income is expected to be constant except for the effect of inflation, this is still the correct method of inputting data.

If you wish to input a special income for the first year or you have estimated the income on a year-by-year basis, the (S)pecified year-by-year option must be selected. If you make an error, the (R)edo all regular income option will zero all variables and allow you to start over again.

The earnings were specified as a constant in the example. Type **C** <CR> and you will see the next question:

```
(M)ONTHLY OR (A)NNUAL INPUT ?
```

Since the annual income was specified as $96,000, select (A)nnual by typing **A** <CR>. The next question is:

```
YEARLY INCOME ? 96000 <CR>
```

The computer now returns to the income menu:

```
INPUT MENU REGULAR INCOME
(C)onstant
(R)edo all regular income
(S)pecified year-by-year
(E)ND CYCLE
Enter Your Choice (C/R/S/E) ?
```

The $96,000 is the only regular income. Select (E)ND by typing **E** <CR>. The next question is:

**VACANCY FACTOR (APPLIED TO ALL VALUES) ?**

This question is designed specifically for the real estate market. If you are making an analysis for other than real estate, or if the vacancy factor is zero, type 0 <CR> as your answer to this question. In this case, the assumption is that there is a five percent vacancy on the average. The value **5**<CR> is typed for five percent vacancy.

The next question is:

```
INPUT CAPITAL INCOME YEAR BY YEAR - ENTER 0,0 TO TERMINATE
Year    Capital Income by Year
?
```

Capital income has different tax consequences than regular income. In real estate, where the tax shelter aspect of the investment is very important, it is necessary to handle carefully each type of income. The only capital income in many real estate investments is the resale dollars at the time the asset is sold. In this case, it is assumed that the investment will be sold at the end of 10 years for $700,000. As your answer to the above question, type the following:

```
  Year       Capital Income by Year
10 <CR>      700000 <CR>
 0 <CR>      0      <CR>
```

Some investors may argue that it is very difficult to estimate resale value at the time of purchase. This is true, but that is no excuse not to perform the task. The analysis technique takes into account all the cash flows associated with an investment. All values must be estimated to perform a complete analysis.

Notice that the **0, 0** sequence was entered to terminate the data entry for the above prompt. The program checks for the paired entry of 0,0 to terminate this phase.

The next question is:

```
INPUT YEARLY COSTS MENU
(C)onstant
(S)pecified year-by-year
(R)edo all yearly costs
(E)ND CYCLE
Enter Your Choice (C/S/R/E) ?
```

This menu is similar to the regular cost menu. In this case we must use both the (C)onstant and (S)pecified year-by-year options. Start by typing **C** <CR> to obtain the following prompt:

```
INPUT YEARLY COST ?
```

The yearly cost is estimated at $25,000. Type **25000** <CR> to return to the menu:

```
INPUT YEARLY COSTS MENU
(C)onstant
(S)pecified year-by-year
(R)edo all yearly costs
(E)ND CYCLE?
Enter Your Choice (C/S/R/E) ?
```

This time type **S** <CR> so you can put in the first year cleanup cost of $15,000. Type **1,15000** <CR> the first time and **0,0** the second time as your answers to the next question:

```
   Year          Amount in ($)
? 1 <CR>      ? 15000 <CR>
? 0 <CR>      ? 0 <CR>
```

The menu again will appear:

```
INPUT YEARLY COSTS MENU
(C)onstant
(S)pecified year-by-year
(R)edo all yearly costs
(E)ND CYCLE?
Enter Your Choice (C/S/R/E) ?
```

This time to (I)ndividual number or (A)ccount

```
                        DEPRECIATION MENU
(L)ine - straight line depreciation
(D)eclining
```

```
(S)um of the year's digits
(R)edo all depreciation calculations
(T)able lookup 1981
(E)ND CYCLE ?
```

The Economic Recovery Act of 1981 changed the way depreciation must be calculated for most business investments. The ability to use the (T)able lookup 1981 method is one of the depreciation selections. All the traditional methods of depreciation are available. Tax laws change constantly, so we tried to make the program flexible enough to take care of future changes.

For the example type **T** <CR> to continue to the next question:

```
YEAR OF EXPENSE (COUNT 0 AS START YEAR) ?
```

The program is designed to add a capital asset at any time during the life of the project. Most of the time, the asset will be purchased at time zero, at the beginning of the investment. Type O<CR> for this example.

The next question is:

```
CAPITAL EXPENSE ?
```

The capital investment was $700,000 for the investment being analyzed. Type **700000** <CR> to continue. The next prompt will ask:

```
SALVAGE VALUE (LAND VALUE) ?
```

Under the 1981 law, you can depreciate an asset to zero salvage value. You may not depreciate the land part of an investment. In this investment, we assumed that the land value was equal to ten percent of the original investment. To input this data, type **700000** <CR>.

The federal government has long used the depreciation laws as a method to direct investors in investing in specific projects. The 1981 law is no exception. The next question tells the story:

```
LOW INCOME HOUSING (Y/N) ? N <CR>
```

You should now be back at the depreciation menu:

```
                        DEPRECIATION MENU
(L)ine - straight line depreciation
(D)eclining
(S)um of the year's digits
(R)edo all depreciation calculations
(T)able lookup 1981
(E)ND CYCLE
Enter Your Choice ?
```

Type **E** <CR> to terminate the input of depreciation and continue with the next question:

```
MORTGAGE MENU
(P)ayments equal
(D)eduction equal
(I)nterest only
(R)edo all mortgages
(E)ND CYCLE?
Enter Your Choice (P/D/I/R/E) ?
```

An equal payment loan or mortgage is the type most individuals see when they contract to purchase a house. Equal reduction or (D)eduction mortgages are very common in land deals where the payments are annual. This type of mortgage is very easy to calculate. (D)eduction was used rather than (R)eduction because the R was used for the (R)edo all mortgages.

An interest only loan is one where the principal is borrowed at the beginning of the loan and only interest is paid until the principal is due at the end of the period of the loan.

In the example, an equal payment loan was assumed at ten percent for twenty-five years. Therefore, answer **P** <CR> to this prompt.

Since an equal payment loan was assumed, the next prompt will appear as:

```
BALLOON YEAR (1000 IF NONE) ?
```

We will not assume that a balloon payment is part of the loan transaction, therefore respond with **1000** <CR> to move to the next prompt.

```
INPUT 0 TO CALCULATE PAYMENT
   YEAR    LOAN AMT ($)   APR (%)   PAYMENTS/YR PAYMENTS    YEARS
? 0 <CR> ? 60000 <CR> ? 10 <CR> ? 12 <CR>   ? 0 <CR> ? 25<CR>
```

When you complete the above you will return to the mortgage menu. This time, enter **E** <CR> to continue to the next menu:

```
***** TAX CATEGORIES *****


(C)orporation
(I)ndividual
Enter Your Choice (C/I)
```

The tax treatment of individuals is very different from that of corporations relative to capital gains. Since it is assumed that the investment is to be a tax shelter, select (I)ndividual by typing **I** <CR>. The next question is:

```
TAX RATE ?
```

Assume you are in the 50 percent tax bracket. Type **50** <CR>. The next menu is:

```
***** OUTPUT MENU ****
(C)RT
(40)-column printer
(80)-column printer
(130)-Column printer?
Enter Your Choice ( C / 40 / 80 / 130 ) ?
```

The larger printer yields a printout that is a little easier to read than the smaller printers. In all cases the information in the printout is the same. Select 80 columns by typing **80** <CR> to make it easier to present the material in this book.

The next prompt relates to the number of years of the study:

```
NUMBER OF YEARS OF STUDY <= 10 ?
```

You may select any study period less than the maximum specified. Since we wish to use the ten-year period type **10** <CR>. The next question is:

```
INTERNAL RATE ?
```

The internal rate of return or yield specified was fifteen percent. Type **15** <CR> to input this information.

The next question is:

```
INFLATION MENU
(S)ingle rate all factors
(I)ndividual rate
REG. COSTS/CAPITAL COSTS ?
```

The program has the ability to input a single inflation rate for the four input factors: regular income, capital income, regular costs, and capital cost. This is the capability we use most of the time. The capability to input individual values for each of the inflation rates is available for those who wish to do so. Selecting (S)ingle rate by typing **S** <CR> results in the following question:

```
INFLATION ?
```

For this run, assume you wish to input a zero inflation rate. Type **0** <CR>.

The results of the analysis will appear on the screen (or on the line printer, depending upon which output option was selected) in a tabular form. Table 24.1 summarizes the results of our example problem.

The first three columns are the data you entered. Column one is the regular income less the five percent vacancy factor. Column two is the capital income. The only capital income in our example occurs in the tenth year, when the asset is sold. The third column contains the regular costs information. Notice that the first year is greater than all the other years. This is because we specified an extra, first-year cleanup cost.

Column 4 contains the capital cost information. This is the only cost that occurred at time zero according to the printout. The model assumes yearend cash flows with the exception of this time zero flow. The capital costs are translated into the depreciation schedule in column 5. These are the costs that resulted from the table lookup method selected for this item.

The next column indicates the mortgage input of cash into the investment. In our example, the only mortgage input was at the beginning of the study period. The mortgage inflow results in the next two columns of output. The first is for interest expense and the second is for reduction of the loan.

The next column is for taxable income. In this case, all taxable income is negative. This means that no taxes are paid except for the capital gains taxes the last year of the investment.

The cash flow column is where the results of all cash is determined. The last column is the adjusted cash flow, which takes into account the tax savings that result from the investment. Notice that this investment's value is as a tax shelter. If the investor was not in the 50 percent tax bracket, the investment would probably be rejected.

The three numbers on the bottom of the sheet summarize the value of the investment. The first number says that the actual cash flow value is negative

TITLE: EXAMPLE PROBLEM, CHAPTER 24 INFLATION 0
DATE : 08/17/83 INTERNAL RATE .15

| YEAR | REGULAR INCOME | CAPITAL INCOME | OTHER COSTS | CAPITAL COSTS | DEPRECIATION EXPENSE | MORTGAGE CASH | INTEREST EXPENSE |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 700000 | 0 | 600000 | 0 |
| 1 | 91200 | 0 | 40000 | 0 | 75600 | 0 | 59744 |
| 2 | 91200 | 0 | 25000 | 0 | 63000 | 0 | 59149 |
| 3 | 91200 | 0 | 25000 | 0 | 56700 | 0 | 58492 |
| 4 | 91200 | 0 | 25000 | 0 | 50400 | 0 | 57766 |
| 5 | 91200 | 0 | 25000 | 0 | 44100 | 0 | 56964 |
| 6 | 91200 | 0 | 25000 | 0 | 37800 | 0 | 56077 |
| 7 | 91200 | 0 | 25000 | 0 | 37800 | 0 | 55099 |
| 8 | 91200 | 0 | 25000 | 0 | 37800 | 0 | 54017 |
| 9 | 91200 | 0 | 25000 | 0 | 37800 | 0 | 52822 |
| 10 | 91200 | 700000 | 25000 | 0 | 259000 | 0 | 51503 |

| YEAR | REDUCTION LOAN | TAXABLE INCOME | TAX RATE | TAX PAID | CASH FLOW | ADJUSTED CASH FLOW |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | -100000 | -100000 |
| 1 | 5682 | -84144 | 0.50 | 0 | -14226 | 27846 |
| 2 | 6277 | -55949 | 0.50 | 0 | 774 | 27848 |
| 3 | 6934 | -48992 | 0.50 | 0 | 774 | 25270 |
| 4 | 7661 | -41966 | 0.50 | 0 | 774 | 21756 |
| 5 | 8463 | -34864 | 0.50 | 0 | 774 | 18205 |
| 6 | 9349 | -27677 | 0.50 | 0 | 774 | 14612 |
| 7 | 10328 | -26699 | 0.50 | 0 | 774 | 14123 |
| 8 | 11409 | -25617 | 0.50 | 0 | 774 | 13582 |
| 9 | 12604 | -24422 | 0.50 | 0 | 774 | 12985 |
| 10 | 521293 | -16803 | 0.50 | 94500 | 98905 | 107306 |

DISCOUNT CASH -84904.50
DISCOUNT TAX SAVINGS -115244.00
DISCOUNT CASH AND TAX SAVINGS 30339.40
*** NOTE: THE AMOUNT SHOWN IN THE LAST ROW OF THE DEPR. COL.
INCLUDES THE RETURN ON CAPITAL.

TABLE 24.1: EXAMPLE PROBLEM WITH A ZERO INFLATION RATE

(at the fifteen percent level for internal rate of return). The second number tells us that the tax savings are substantial. In this case, a negative number means good tax savings. The next number is the sum of the two previous numbers. The net present value of the investment, considering both cash flow and tax savings, is $30,339.45. This means the rate of return is greater than fifteen percent.

The next question you will see after the computer has printed its results is:

```
COMMENTS FOR PRINTOUT...?
```

This question will be repeated four times. The objective of these four questions is to allow you to add some notes to the printout. If you do not care to add any comments, simply press <CR> each time the prompt appears.

The next menu that will appear is:

```
RECYCLE MENU
(R)egular income
(C)apital income
(Y)early expenses
(D)epreciation expense
(M)ortgages and loans
(I)nflation and output options
(E)nd program
          Enter Your Choice (R/C/Y/D/M/I/E) ?
```

This menu allows you to change any of the input factors without changing anything else. Assume the only change you wish to make is to change the inflation rate from 0 to 6 percent. Type **I** <CR> as your answer to the above question. The next series of questions and answers are shown:

```
(C)orporation
(I)ndividual
? I <CR>
TAX  RATE  ? 50 <CR>

OUTPUT MENU
(C)RT
(40)-column printer
(80)-column printer
(130)-column printer? 80 <CR>
NUMBER OF YEARS OF STUDY <= 10 ? 10 <CR>
INTERNAL RATE ? 15 <CR>

INFLATION MENU
(S)ingle rate all factors
(I)ndividual rate
REG. COSTS/CAPITAL COSTS? S <CR>
INFLATION? 6 <CR>
```

The program will now re-evaluate the same problem but with the inflation factor of 6 percent applied to all appropriate cash flows. The results of this analysis are shown in Table 24.2.

The effect of inflation on the analysis is interesting to examine. First, notice that the net present value increased from $30,339.45 to $190,570.50. The original selling price, estimated to be $700,000, is now $1,253,593. The regular income increased from $91,200 to $163,325 in the last year. Costs also increased, but the depreciation expense did not change, and the mortgage costs remain constant. This behavior is reflected in real life.

Another interesting aspect of inflation is that the value of the investment shifted from a tax shelter to a positive cash flow investment. In the sixth year, with only a 6 percent inflation rate, this investment starts to pay taxes.

The analysis is complete and detailed. It will help you understand and predict the consequence of business decisions. When you first start using it, compare it with the old decision-making method you have been using. Apply

TITLE: EXAMPLE PROBLEM, CHAPTER 24 INFLATION 6
DATE: 08/17/83 INTERNAL RATE .15

| YEAR | REGULAR INCOME | CAPITAL INCOME | OTHER COSTS | CAPITAL COSTS | DEPRECIATION EXPENSE | MORTGAGE CASH | INTEREST EXPENSE |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 700000 | 0 | 600000 | 0 |
| 1 | 96672 | 0 | 42400 | 0 | 75600 | 0 | 59744 |
| 2 | 102472 | 0 | 28090 | 0 | 63000 | 0 | 59149 |
| 3 | 108621 | 0 | 29775 | 0 | 56700 | 0 | 58492 |
| 4 | 115138 | 0 | 31562 | 0 | 50400 | 0 | 57766 |
| 5 | 122046 | 0 | 33456 | 0 | 44100 | 0 | 56964 |
| 6 | 129369 | 0 | 35463 | 0 | 37800 | 0 | 56077 |
| 7 | 137131 | 0 | 37591 | 0 | 37800 | 0 | 55099 |
| 8 | 145359 | 0 | 39846 | 0 | 37800 | 0 | 54017 |
| 9 | 154080 | 0 | 42237 | 0 | 37800 | 0 | 52822 |
| 10 | 163325 | 1253590 | 44771 | 0 | 259000 | 0 | 51503 |

| YEAR | REDUCTION LOAN | TAXABLE INCOME | TAX RATE | TAX PAID | CASH FLOW | ADJUSTED CASH FLOW |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | -100000 | -100000 |
| 1 | 5682 | -81072 | 0.50 | 0 | -11154 | 29382 |
| 2 | 6277 | -47767 | 0.50 | 0 | 8956 | 32839 |
| 3 | 6934 | -36347 | 0.50 | 0 | 13419 | 31592 |
| 4 | 7661 | -24590 | 0.50 | 0 | 18150 | 30444 |
| 5 | 8463 | -12473 | 0.50 | 0 | 23164 | 29401 |
| 6 | 9349 | 28 | 0.50 | 0 | 28465 | 28465 |
| 7 | 10328 | 6642 | 0.50 | 3321 | 30793 | 30793 |
| 8 | 11409 | 13696 | 0.50 | 6848 | 33238 | 33238 |
| 9 | 12604 | 21221 | 0.50 | 10611 | 35806 | 35806 |
| 10 | 521293 | 35551 | 0.50 | 222994 | 576357 | 576357 |

DISCOUNT CASH 115183.00
DISCOUNT TAX SAVINGS -75387.80
DISCOUNT CASH AND TAX SAVINGS 190570.50

*** NOTE: THE AMOUNT SHOWN IN THE LAST ROW OF THE DEPR. COL.
INCLUDES THE RETURN ON CAPITAL.
TABLE 24.2: EXAMPLE PROBLEM WITH A 6 PERCENT INFLATION RATE

it to decisions you have made in the past. After you learn how to use it, start applying it to your business with care. Be sure to check the input data each time you use the system to make sure the correct data has been entered. A data entry error or combinations of data entry errors may result in an erroneous decision that may cost the decision-maker a substantial sum of money. Therefore, care should be taken in the collection of the data, and thoughtful consideration must be given to the pattern of projections that apply to the future cash flows.

# 25. (F)orecast

This program analyzes a set of data organized in a file consisting of a set of matched points X and F(x). The product of the analysis is: a least squares, first degree regression equation; a correlation coefficient; a plot of the points with or without a regression line; or a forecasted value, depending on the menu selection made. The plot may have prediction and confidence limits drawn as well as the regression line.

The program has been designed to be used by persons with limited computer experience. However, it is strongly suggested that the potential user perform some individual research into the basics of linear regression and correlation in order to interpret the output of the program correctly. The visual nature of the analysis is designed to aid the user in drawing conclusions from the results. A nonmathematical review of the theory used to develop the analysis is given. A discussion of the meaning of confidence limits, prediction limits, and confidence levels is included for the user with a statistics background.

## 25.1 Running The Program

The program requires Microsoft MBASIC to operate under control of the Apple CP/M operating system. Printer output is designed so that it may be used with almost any printer, including a daisy wheel. If you are equipped with an MX-80 or similar printer, you will experience no problem in producing printed output of the screen plots.

The authors have purposely written the program with a simple screen plotting technique that avoids special-purpose graphics packages. While slow, the screen and printer "graphics" subroutines use only the standard ASCII character set common to all microcomputers sold in the United States.

To run FORECAST from Apple CP/M simply type **MBASIC FORECAST** followed by a <CR>. It is assumed that the program diskette is in drive A and a data diskette is in drive B. The program provides the user ability to store data for future use. It is highly suggested that you do not store data on the program diskette. Instead, create a formatted Apple CP/M diskette and copy the system using the /S option under COPY. As is standard practice, the data diskette goes in drive B.

As soon as the system is up, the introductory screen promo will appear as:

```
*** APPLE CP/M REGRESSION ANALYSIS ***
                by
      STEVEN M. ZIMMERMAN, Ph.D.
              LEO CONRAD
      DONALD R. SMITH Ph.D., P. E.

      Press < CR > to Begin ?
```

The next screen will produce the following:

```
REGRESSION WITH PREDICTION & CONFIDENCE LIMITS
LEAST SQUARES APPROACH
f(x) = a + bX

*** WARNING: USE CODED DATA WHENEVER POSSIBLE.
DETAILS ( Y/N) ?
```

Answer **Y** so as to examine the warning we have built into the program the following:

```
Regression analysis requires the use of some very large numbers and
then finding the difference between these large numbers. It is good
practice to use coded data whenever possible.
```

The problem of value rounding has been encountered many times when doing regression analysis on microcomputers. In regression analysis, because of the need to add the sums, the squares, and the product of numbers, and then find the difference between these large numbers, rounding errors are critical. One way to avoid this problem is to code the data so that the original data are smaller numbers. This coding reduces potential errors.

When working with large numbers or long numbers with many significant figures, be sure to look at the pictures drawn by the program. If the regression line and the limits ever look like they do not fit the scatter diagram, you most likely have encountered a rounding error, and it will be necessary to code your data. Under normal circumstances, you should not encounter any difficulty in using the program without coding.

The next prompt is:

```
NUMBER OF DATA POINTS ?
```

The amount of data points that may be handled by this program is a function of the amount of memory in your computer. We were able to specify over 300 data points in our 64K system without obtaining an out of memory error message.

For the sample run, we used 25 for the sample problem. Type **25** <CR>, and you will see:

```
*** DATA INPUT MENU ***
(D)isk
(K)eyboard
(R)ead
Enter Your choice ( D/K/R ) ?
```

Three methods of data input are available in this program. The first is by way of a file kept on a diskette. If you have previously entered data and stored the data set on diskette, you would select the "D" option to retrieve the data.

Disk files do not come into existence by themselves. To create a disk file, the data must be input into the computer. The program is designed to allow keyboard input of data that then may be saved as a disk file for future update and use. We will illustrate the results when the (K)eyboard option is selected.

The third option is (R)ead. In this option, the computer looks for data added to the BASIC program in DATA statements. Line number 1890 of the

program is a DATA statement that contains data to be used only as an example. You may use this data by selecting the (R)ead option. You may replace this line with your own data statement by simply typing 1890 DATA and your own data. Take care to set up the data to match the needs of the program. The data is listed in X,Y or X,F(X) pairs. At the end of the data is a matched pair of points carrying the values 9999,9999. This pair of values is used to tell the computer you have completed the task of inputting your data.

To illustrate how FORECAST operates, we will consider the following example data:

| Data No. | x | F(x) |
|---|---|---|
| 1 | 1.1 | 100 |
| 2 | 1.3 | 150 |
| 3 | 1.2 | 134 |
| 4 | 1.5 | 200 |
| 5 | 1.6 | 220 |
| 6 | 1.4 | 178 |
| 7 | 1.7 | 240 |
| 8 | 1.8 | 255 |
| 9 | 1.8 | 265 |
| 10 | 1.9 | 300 |

with data statements, it would look like the following:

```
1890 DATA 1.1,100,1.3,150,1.2,134,1.5,200,1.6,220,1.4,178
1892 DATA 1.7,240,1.8,255,1.8,265,1.9,300,9999,9999
```

The assumption is that you typed **K** <CR> as your answer to the menu question. If you wish to input the same data using the keyboard, the prompt and responses would be:

```
                    *** DATA INPUT SCREEN ***
                                                  9999 <CR> 9999 <CR>
ELEMENT NO      X-VALUE       FUNCTION OF X            TO STOP
    1             1.1             100
    2             1.3             150
    3             1.2             134
    4             1.5             200
    5             1.6             220
    6             1.4             178
    7             1.7             240
    8             1.8             255
    9             1.8             265
   10             1.9             300
   11            9999            9999
```

The values 9999, 9999 are used here and in data statements as the method of telling the computer you have completed the inputting data. This communication method is also true in the manner in which the disk data files have been structured.

NOTE: In the above data input, a <CR> is necessary after the keying-in of each data element.

Once you have completed the input of data from either keyboard or DATA statements, the following output menu appears:

```
OUTPUT MENU
(A)dd values (C)hange (D)elete (E)nd (F)ile-disk (I)nsert (L)ist
  (M)onitor (P)rinter (X)merge data files
                        Enter your choice ?
```

The output menu is one of the main menus of the program. You have the option in this menu of maintaining the data file in almost any manner you wish. You may add data, change values, delete values, and insert data. After you have corrected the data as needed, you may select (F)ile-disk to save the data for future use.

In addition to the abilities noted above, the program allows you to merge two data files into a single file. The (X)Merge option performs this task. After you have merged the files as desired, you then may save the results for future use.

Additionally, you are able to list your data for examination and proceed with the remaining abilities of the program using either the monitor or printer.

The reason we selected the names of the options as shown was to allow the first letter to identify the option in the maximum number of cases. You may select an option by typing the letter in the parentheses or by typing the entire word. As long as the first letter on a line matches the one in parentheses (upper or lowercase), you will get the desired results.

Review each option. The first option to be examined is (L)ist, since it is convenient to see what the data file now looks like and the effect of your actions on the file. Type **L** <CR> and you will see:

| I | X | F(X) | I | X | F(X) |
|---|---|---|---|---|---|
| 1 | 1.1 | 100 | 2 | 1.3 | 150 |
| 3 | 1.2 | 134 | 4 | 1.5 | 200 |
| 5 | 1.6 | 220 | 6 | 1.4 | 178 |
| 7 | 1.7 | 240 | 8 | 1.8 | 255 |
| 9 | 1.8 | 265 | 10 | 1.9 | 300 |

```
                        OUTPUT MENU
(A)dd values (C)hange (D)elete (E)nd (F)ile-disk (I)nsert (L)ist
(M)onitor (P)rinter (X)merge data files
                        Enter your choice ?
```

The next option to examine is the (A)dd option. To select this option, type **A** <CR>, and the following will appear:

```
                    *** DATA INPUT SCREEN ***
                                          9999 <CR> 9999 <CR>
ELEMENT NO      X-VALUE      FUNCTION OF X        TO STOP
    11          ?
```

To continue, answer 2.2 <CR> 320 <CR> for sample 11; **2.3** <**CR**> **333** <**CR**> for sample 12, and **9999** <**CR**> **9999** <**CR**> for sample 13. After completing this task, your program returns to the main menu. Again the (L)ist option is selected to see what has happened to your data set. The following results:

```
 I          X          F(X)          I          X          F(X)
 1          1.1        100           2          1.3        150
 3          1.2        134           4          1.5        200
 5          1.6        220           6          1.4        178
 7          1.7        240           8          1.8        255
 9          1.8        265           10         1.9        300
 11         2.2        320           12         2.3        333
                          OUTPUT MENU
(A)dd values (C)hange (D)elete (E)nd (F)ile-disk (I)nsert (L)ist
  (M)onitor (P)rinter (X)merge data files
                        Enter your choice ?
```

The next option to be examined is the (F)ile-disk option, to save the above data for future use. We recommend that you save your data files often and on two different disks for backup. The task of typing data is not much fun, and you will want to avoid doing it a second time, as you would if there were a power surge or power outage in the middle of a typing session. Typing **F** <CR> results in the following prompt:

```
DATA FILE NAME: ?
DISK (A,B,C,D) ?
```

Apple CP/M requires you to identify the disk drive where a file is to be stored or found, unless you are using the default drive. It is good practice always to identify the disk drive. To answer the above question, you must type d:filespec, where d is the disk drive where the file is to be saved and filespec is the name of your file. Type **B:REGDATA** <**CR**>. On your Apple you may use upper or lowercase letters in naming a file. The system automatically assumes you used uppercase letters and stores the file in this manner. You may recall a file in the same manner. There is no difference to the computer (in this case) between REGDATA and regdata, for example.

After you have completed the input of the disk filename, you will see the light on the specified disk drive glow, hear the drive go into operation, and when the noise stops and the light dims, the file will be saved. If the drive door is open, or if you do not have a diskette in the drive, you will get an error message.

When the computer has completed the task of saving the file, the program returns to the main menu. Now that the file has been saved, all sorts of operations may be performed on the file in RAM without affecting the file on the diskette. First, delete data point 9. The prompts, responses, and listing of the program that result are:

```
                          OUTPUT MENU
(A)dd values (C)hange (D)elete (E)nd (F)ile-disk (I)nsert (L)ist
  (M)onitor (P)rinter (X)merge data files
                    Enter your choice ? D <CR>
Number to be deleted (0 to exit) ? 9 <CR>
                          OUTPUT MENU
(A)dd values (C)hange (D)elete (E)nd (F)ile-disk (I)nsert (L)ist
  (M)onitor (P)rinter (X)merge data files
                    Enter your choice ? L <CR>
```

```
 I          X          F(X)          I          X          F(X)
 1          1.1         100          2          1.3         150
 3          1.2         134          4          1.5         200
 5          1.6         220          6          1.4         178
 7          1.7         240          8          1.8         255
 9          1.9         300          10         2.2         320
 11         2.3         333          12         9999        9999
                          OUTPUT MENU
(A)dd values (C)hange (D)elete (E)nd (F)ile-disk (I)nsert (L)ist
  (M)onitor (P)rinter (X)merge data files
                        Enter your choice ?
```

Notice the former sample 9 is gone and all samples following 9 are moved up one. Sample 12 is now 9999,9999, the terminating sample. This sample is printed only when an odd number of samples are used. To aid in identifying the answers, a <CR> has been typed behind each one.

The next option to be examined is the (I)nsert option. The values 1.87,278 will be inserted as sample number 9. In regression analysis the order in which data is listed is not important in the analysis but it may be of value to the people evaluating the situation to have the data in some specific order. The prompts and responses needed to insert the data and a new listing of the data are:

```
                          OUTPUT MENU
(A)dd values (C)hange (D)elete (E)nd (F)ile-disk (I)nsert (L)ist
  (M)onitor (P)rinter (X)merge data files
                    Enter your choice ? I <CR>
Number to be inserted (0 to exit) ? 9<CR>
X and F(X) ? 1.87,278 <CR>
                          OUTPUT MENU
(A)dd values (C)hange (D)elete (E)nd (F)ile-disk (I)nsert (L)ist
  (M)onitor (P)rinter (X)merge data files
                    Enter your choice ? L <CR>
 I          X          F(X)          I          X          F(X)
 1          1.1         100          2          1.3         150
 3          1.2         134          4          1.5         200
 5          1.6         220          6          1.4         178
 7          1.7         240          8          1.8         255
 9          1.87        278          10         1.9         300
 10         2.2         320          11         2.3         333
                          OUTPUT MENU
(A)dd values (C)hange (D)elete (E)nd (F)ile-disk (I)nsert (L)ist
  (M)onitor (P)rinter (X)merge data files
                        Enter your choice ?
```

The new data point has been inserted as desired.

The next option to be examined is (X)Merge. For illustration only, merge B:REGDATA to the above file. The prompts responses and a listing of the results appear as follows:

```
                          OUTPUT MENU
(A)dd values (C)hange (D)elete (E)nd (F)ile-disk (I)nsert (L)ist
  (M)onitor (P)rinter (X)merge data files
```

```
                    Enter your choice ? X <CR>
INPUT Disk (A,B,C,D):FILE ? B:REGDATA <CR>
                          OUTPUT MENU
(A)dd values (C)hange (D)elete (E)nd (F)ile-disk (I)nsert (L)ist
  (M)onitor (P)rinter (X)merge data files
                    Enter your choice ? L <CR>
```

| I | X | F(X) | I | X | F(X) |
|---|---|---|---|---|---|
| 1 | 1.1 | 100 | 2 | 1.3 | 150 |
| 3 | 1.2 | 134 | 4 | 1.5 | 200 |
| 5 | 1.6 | 220 | 6 | 1.4 | 178 |
| 7 | 1.7 | 240 | 8 | 1.8 | 255 |
| 9 | 1.87 | 278 | 10 | 1.9 | 300 |
| 11 | 2.2 | 320 | 12 | 2.3 | 333 |
| 13 | 1.1 | 100 | 14 | 1.3 | 150 |
| 15 | 1.2 | 134 | 16 | 1.5 | 200 |
| 17 | 1.6 | 220 | 18 | 1.4 | 178 |
| 19 | 1.7 | 240 | 20 | 1.8 | 255 |
| 21 | 1.8 | 265 | 22 | 1.9 | 300 |
| 23 | 2.2 | 320 | 24 | 2.3 | 333 |

```
                          OUTPUT MENU
(A)dd values (C)hange (D)elete (E)nd (F)ile-disk (I)nsert (L)ist
  (M)onitor (P)rinter (X)merge data files
                      Enter your choice ?
```

An examination of the data added will show that these numbers correspond to the ones that were saved when the (F)ile-disk option was examined. This file is of little value for analysis, so perform one more option with it before starting over. Change line 11 to 99,666. To obtain this result the following steps are performed:

```
                          OUTPUT MENU
(A)dd values (C)hange (D)elete (E)nd (F)ile-disk (I)nsert (L)ist
  (M)onitor (P)rinter (X)merge data files
                    Enter your choice ? C <CR>
Number to change ? 11 <CR>
X AND F(X) ? 99,666 <CR>

                          OUTPUT MENU
(A)dd values (C)hange (D)elete (E)nd (F)ile-disk (I)nsert (L)ist
  (M)onitor (P)rinter (X)merge data files
                    Enter your choice ? L <CR>
```

| I | X | F(X) | I | X | F(X) |
|---|---|---|---|---|---|
| 1 | 1.1 | 100 | 2 | 1.3 | 150 |
| 3 | 1.2 | 134 | 4 | 1.5 | 200 |
| 5 | 1.6 | 220 | 6 | 1.4 | 178 |
| 7 | 1.7 | 240 | 8 | 1.8 | 255 |
| 9 | 1.87 | 278 | 10 | 1.9 | 300 |
| 11 | 99 | 666 | 12 | 2.3 | 333 |
| 13 | 1.1 | 100 | 14 | 1.3 | 150 |
| 15 | 1.2 | 134 | 16 | 1.5 | 200 |
| 17 | 1.6 | 220 | 18 | 1.4 | 178 |
| 19 | 1.7 | 240 | 20 | 1.8 | 255 |
| 21 | 1.8 | 265 | 22 | 1.9 | 300 |
| 23 | 2.2 | 320 | 24 | 2.3 | 333 |

```
                    OUTPUT MENU
(A)dd values (C)hange (D)elete (E)nd (F)ile-disk (I)nsert (L)ist
  (M)onitor (P)rinter (X)merge data files
                  Enter your choice ?
```

This completes all the options in the main menu with the exception of (M)onitor and (P)rinter, which relate to how the data will be presented when analyzed. There are special graphics used for each of the two options. In the (M)onitor option, all output is to the monitor. In the (P)rinter option, you will get printer output only. All symbols that would have appeared on the screen are reproducible on any printer, including a daisy wheel.

Before performing the analysis, select the (E)nd option, forget about the data now in the computer, and return to the data set saved as REGDATA. **RUN** <CR> is entered, the credits appear, and you should answer the size question with **2d**. When the question:

```
                *** DATA INPUT MENU ***
(D)isk
(K)eyboard
(R)ead
Enter your choice ( D/K/R ) ?
is asked, type D <CR> with the following results:
DATAFILE NAME = ?
DISK DRIVE (A/B/C/D) = ?
```

The above question is answered with **REGDATA** <CR>, and **B** <CR>. After the last <CR>, you will be returned to the main menu. The first option to be reviewed is the printer option. Type **P** <CR> and you will hear the printer start to work. The following output will appear on your printer:

```
REGRESSION WITH PREDICTION LIMITS LEAST SQUARES APPROACH
f(x) = a + b * x
```

In the program, A is the intercept, and B is the slope. The entire equation will be written to aid you in identifying which is which.

The next screen prompt is:

```
TITLE ?
```

One objective in printer output is to include identification of the problem at all times. It is of no value to find a series of computer runs with all sorts of answers and no identification to help establish what the question was.

To the above question, answer **REGDATA** to identify the data being analyzed with its file name. The next two identifcation questions, and the answers given are:

```
DATE (MM/DD/YR) ? 08/30/83 <CR>
OPERATOR ? D. R. SMITH <CR>
```

The date was input in the above manner in order to avoid the need for a comma. A comma is a separator and may not be used in a regular INPUT statement, as used to input this information.

The three items will be printed as they are input:

```
REGDATA
08/30/82
OPERATOR: Conrad
```

followed by the reproduction of the data, equation, and correlation coefficient:

| I | X | F(X) | I | X | F(X) |
|---|---|---|---|---|---|
| 1 | 1.1 | 100 | 2 | 1.3 | 150 |
| 3 | 1.2 | 134 | 4 | 1.5 | 200 |
| 5 | 1.6 | 220 | 6 | 1.4 | 178 |
| 7 | 1.7 | 240 | 8 | 1.8 | 255 |
| 9 | 1.8 | 265 | 10 | 1.9 | 300 |
| 11 | 2.3 | 319 | 12 | 2.3 | 333 |

```
F(X) =  -99.18934  +  192.2266 * X
Correlation Coefficient: 0.9856950
```

The regression equation says the intercept is -99.1893 and the slope is 192.2266. For many applications, this is all the information required. The correlation coefficient of 98 percent (.98569505) means that the data is close to a straight line. A correlation coefficient of +1 is a perfect straight line with a positive slope, while a correlation coefficent of -1 is a straight line with a negative slope, and an answer of 0 means the data does not possess any linear characteristics.

Correlation coefficients in the neighborhood of 80 percent or above add to your confidence in using the forecast based on the regression equation. Correlation coefficents in the neighborhood of 50 percent or less say you are on very dangerous ground when using the regression equation. Results between 50 percent and 80 percent tell you to proceed with caution. The correlation coefficient is simply one more number that helps make the forecast and guides you in what confidence should be placed on the forecast.

To use the regression equation for forecasting, you need only know the value of X. If X is a time variable, you must be careful not to project your results too far into the future. Assume you wish to know the value of the equation when X is 1. You may substitute the value into the equation or use the (F)orecast option in the following menu:

```
                    ACTION MENU
      (P)lot (F)orecast (N)ew run (O)utput (M)enu (E)nd
                  Enter your choice ?
Typing F <CR> results in the following screen prompt:
                 FORECAST DATA OPTION INPUT
Input the desired X value for prediction and the desired predic-
tion limits (1/2/3)
PREDICTED X VALUE      PREDICTION LIMITS (1/2/3)      0 0 TO STOP
?
```

The program includes the ability to calculate the value of the regression equation, the values of the confidence limits, and the values of the prediction limits. Confidence limits help tell you how good the estimate of the regression line is. You will be able to see that the confidence limits are very narrow near the center of the data and how they widen out toward the end of the data set. If you project beyond the data set, confidence limits will become wider and

wider. Although not a perfect guide, the confidence limits tell you a little more about the estimate you are developing.

Prediction limits are similar to confidence limits in that they are very tight near the center of the data set and widen near the ends of the data. Prediction limits pertain to the individual data points. If you are concerned with individual occurrences, the prediction limits should be used. The (F)orecast option will calculate specific values for you, while the (P)lot option will draw the prediction and confidence limits around the regression line as the line is fitted to the data.

In both confidence and prediction limits, you must specify the level at which the limits are to be calculated. Common usage says two sigma limits (95 percent confidence level) or three sigma limits (99+ percent confidence level) should be used. We will review this idea in more depth in the theory section. For this trial run assume you wish to use three sigma limits.

Assume you answered **1,3**. The printer will produce the following results:

```
f(X) = 97.0366
Lower Confidence Limit= 26.58265
Upper Confidence Limit= 175.4783
Lower Prediction Limit=-46.28552
Upper Prediction Limit= 248.3465
Beyond range of data ********
```

Since you selected a value of X beyond the limits of the data set, the computer added the warning shown above. Statisticians consider projecting results beyond the range of data used in regression analysis dangerous. The program is designed to tell you when this has happened. In many business situations, projection of results into the future is one of the main (but somewhat overused) objectives of regression analysis.

After you have completed all the analytical forecasting, you may terminate the session by typing **0,0** <CR>. You will now see the action menu:

```
                    ACTION MENU
(P)lot (F)orecast (N)ew run (O)utput (M)enu (E)nd
                Enter your choice ?
```

This time, select **P** for (P)lot and examine the printer graphics available in the program. After selecting this option you will be asked to select one of the following:

```
PLOTTING OPTIONS ARE:
    (1) REGRESSION LINE f(x) = a + bx
    (2) DATA points only
                Enter your choice ( 1/2 ) ?
```

There is often a need to produce a scatter diagram of your data without the inclusion of a regression line or limits. The second option yields this result. The first option assumes you wish to plot both a regression line and confidence limits, prediction limits, or both.

Type **1** <CR> to select the first option. The next question is:

```
DESIRED SIGMA LIMITS (1,2,3) = ?
```

When you calculated the confidence and prediction limits it was necessary to specify a confidence level. This is also true for the task of plotting the limits. When using printer graphics, it is suggested that you select three sigma limits to get a good usable picture. When using the monitor option, which provides a form of screen graphics, tighter limits may be selected with good pictorial results.

For the sample run, type **3** <CR> and proceed to the next question:

```
ANALYSIS TYPE: (B)oth,(C)onfidence interval, (P)rediction
interval ?
```

Because of the limited graphics available with printer graphics, it is not recommended that you try plotting both confidence and prediction limits at the same time. For the sample run, make use of C one time, and then repeat using P. Figure 25-1 illustrates the printer output using the **C** option.

The next question is:

```
(L)ow point or (O)rigin ?
```

You may plot the graph from the point of the lowest data point or the origin. Be careful when using the origin with time data. If you input your date 19xx, the origin is the year 0. A plot starting in year 0 to the present time of the sales of microcomputers would be of little value.

Input **0** <CR> to use the origin for the example and proceed to the next prompt:

```
(H)igh point or (P)rojection ?
```

Since the objective is to project results into the future, then the (P)rojection option is often desirable. Since the limits, the regresssion line, and the data are plotting a projection simultaneously, the user is provided an additional degree of understanding of the process being used. Select (H)igh for the sample run to see what happens and then return to use the (P)rojection option.

Press <CR> and select (O)utput in the action menu to return to the output menu. This time select (M)onitor and repeat what you have just performed to examine screen display on the Apple. The screen display will be similar to Figures 25.1 and 25.2.

## 25.2 Theory

The initial assumption in all statistical analysis is that the past in some way reflects the future and that by examination of the past you will be able to draw conclusions about the future. If the process being studied cannot be projected into the future for whatever reason, then no computer program will help you obtain useful information. Only you can decide on the strength of the past and its potential relationship to future events.

Statistics will help you draw better conclusions by examining the mathematical nature of the process and its stability or instability for the purpose of prediction. Careful use of this program provides warning signs to aid you in determining when you may be in trouble, by using a regression equation to

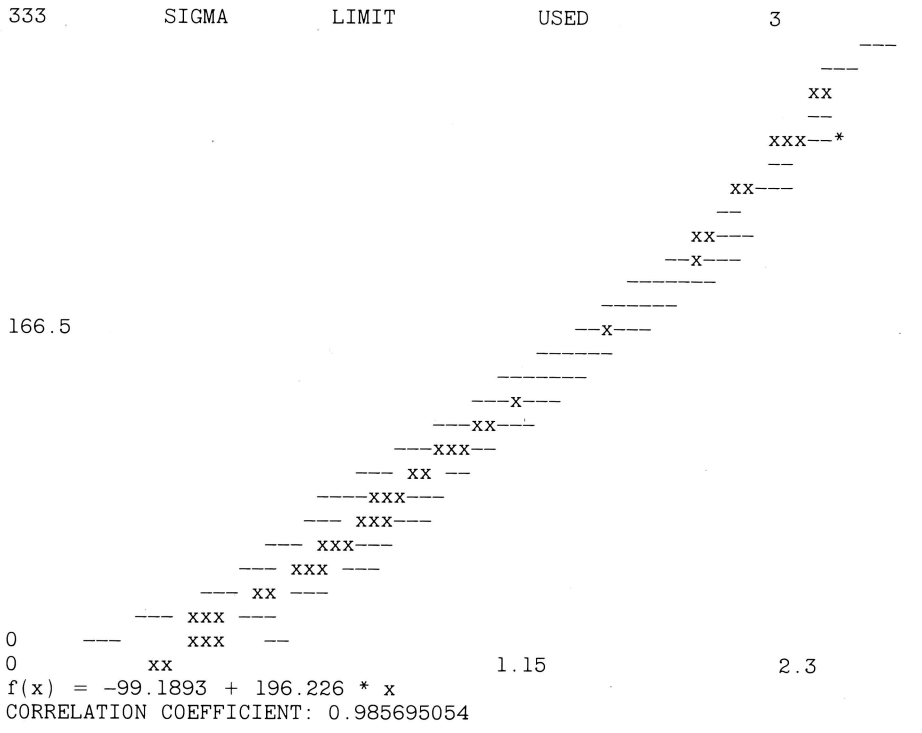



**Figure 25.1: Plot of Example Data Using the "C" option**

make a decision about a process. In the final analysis, people make decisions, and computer programs or calculators can only aid in making the decision more accurately or faster.

In statistics, the use of confidence is fundamental to the drawing of conclusions about a set of data. Statistics is a betting situation. By using two sigma

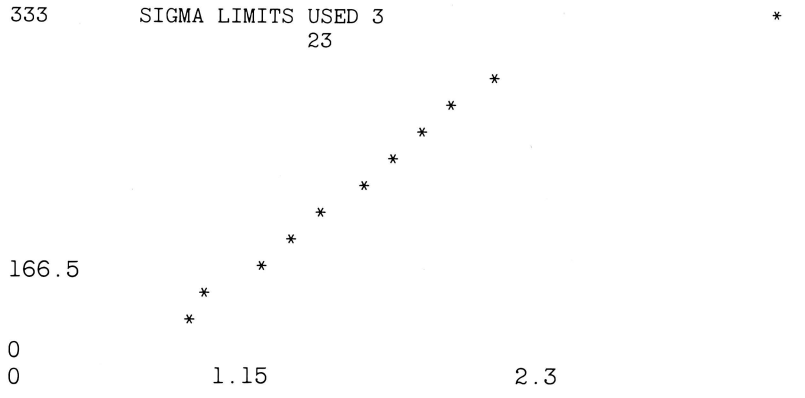



**Figure 25.2: Scatter Diagram of the Example Data (Data Points Only)**

limits, you are using the odds of 1/20 that, by chance alone, a result will occur beyond the limits when nothing has occurred to produce poor results. The use of three sigma limits means that you have selected odds of 5.2 to 1000. If something occurs beyond the three sigma limits, there is a very good chance that an outside influence has made itself felt in the process.

What odds should you select? There is no simple answer to this question. It's your money or your job that is on the line. To make an investment or to do nothing are two decisions, and either can be wrong. The program is not a substitute for good judgment. It helps you make good judgments. You still must take the risk and make the decision. It is suggested that you use two or three sigma limits as your guide. Common practice in industry is to use mostly three sigma limits.

Prediction and confidence limits have not found extensive application in the past due to the difficulty in calculation. This program eliminates that problem. Use the program at first as you would any simple linear regression analysis program. Look at the limits and the data points. Remember, the individual points are related to the prediction limits, and the trend line is related to the confidence limits. If the individual data points fall outside the prediction limits, expect the correlation coefficient to be near zero. If all the individual data points fall within the prediction limits, the correlation coefficient should be near one or negative one. Both the correlation coefficient and the use of prediction and confidence limits aid you in understanding how well the regression line fits the data available and nothing more.

Least squares is the theoretical method used to fit the regression line to the data. The concept is to minimize the distance from each point to the line. If this were done considering the direction of the difference, the results would be zero and would create some mathematical difficulty in use. The choice is to use the absolute value of the difference or the square of the distance to get rid of the sign. Since the math is easier when using the square, this is commonly the choice made.

The least squares of the distance from each point to the line is minimized when the line indicated is used. The computer produces this result without your getting into the detailed calculation procedures.

## 25.3 Examining The Program

The first line of the program is the program identification used to aid in saving the program as noted in earlier discussions. Line 20 establishes the width of the output as 80. Normally this is not necessary, but if you break the program while it is in 40-column mode, the output will remain in 40-column mode, and that was not wanted. Line 20 clears some space for string variables.

Lines 40 through 110 are a subroutine used to print the warning about coded data. This section of the program consists primarily of simple print statements used to display the warning on the screen.

The program headings are printed in lines 160 through 230. In addition, in line 210 the data amount counter is set equal to zero to begin the program.

The first question is in line 230. This question requests information about the maximum amount of data storage to be allocated in the program. In the next line, the DIMensions of the variables X and F are set according to the instructions obtained in line 250. Line 245 contains a DIM statement that is used to place values on the screen when the plot option is selected.

Line 270 is the data input statement. The appropriate logic checking is conducted in lines 300 through 330. You are given the option in line 270 of inputting from disk files or the keyboard, or using the read DATA statement procedure.

Line 280 routes the flow of the program to input the disk drive location of the disk file and its name. Then it opens the file if this option was selected.

Line 290 starts the counter that counts the number of data points input by any method. If input is from the read statement, line 300 is executed. If input is from the disk file, line 310 is executed, and if the input is from keyboard, then line 320 is executed.

Line 330 checks to see if the "9999" (end-of-data input) sequence has been keyed in. If so, the program transfers to 340; if not, the transfer is to line 290 where the data counter is increased by one.

Line 350 sets N equal to I-1, which is the number of data points that were input. This value of N as the number of data points is carried throughout the program. Line 360 closes the disk file whether one was open or not, and then directs the flow of the program to the output menu in line 440. For this part of the program, you will be returned to line 440 again and again after each routine is complete.

Starting in line 470, ten options are available. Lines 470 through 600 direct the flow of the program depending on the option selected. Each of the routines will be examined in turn.

The first option in line 470 is the END option. If this option is selected, the program terminates. To restart, simply type **RUN** from the keyboard.

Line 480 is the LIST option and if selected, program control is transferred to line 370.

The CHANGE option is in line 490 with control transferred to line 440.

Line 500 is the ADD option which allows additional data to be added to the existing data set. Logic control is transferred to line 290.

The MONITOR option provides for all data displays to be routed to the screen. This option is checked in line 510 with program control routing to line 710.

The PRINT option is checked in line 520 with control routed to line 710.

Deleting data is checked in line 530 with program control transferred to line 620.

The INSERT data option is in line 540 with program control branching to line 660.

The final option, MERGE files, is checked in line 550 and a branch to line 700 is conducted.

Lines 710 through 850 perform most of the numerical calculations necessary to determine the regression equation. The major variables used in the calculations are summarized below:

| Variable | Use |
|---|---|
| SA# | sum of the $y^2$ |
| SX# | sum of the x's |
| SY# | sum of the y's |
| SC# | sum of the product x * y |
| S2# | sum of the $x^2$ |
| S3# | sum of the $x^3$ |
| S4# | sum of the $x^4$ |
| S5# | sum of the $x^2$ * y |

Notice that all the variables have a # sign behind them. All calculations are performed in double precision to increase the accuracy of the results. Even with this approach, rounding errors are possible, as indicated in earlier discussions.

In line 860, the print option is checked. If line printing was desired, the output is managed in lines 870 through 930.

Lines 940 through 970 perform additional calculations needed to compute the slope and intercept of the regression equation.

Lines 1020 and 1030 are the action menu. This is the second main menu of the program. From this menu, you may be directed to plot results on screen and printer, forecast, start a new run, return to the output menu, or terminate the program.

If the (P)lot option is selected, line 1050 directs you to choose whether you want a plot with or without a regression line, and then sends the program to line 1120. In line 1120, you must select the low point from which the plot will start, and in line 1140 you must select the high point.

The screen or printer plotting is managed in lines 1170 through 1710. By the nature of Apple CP/M, you can have a screen display or a printer display with the CRT disabled. It is suggested that you use the monitor option first, then return for the printer option in the action menu.

To accomplish a generalized screen plotting of the data, the authors resorted to a basic technique that defines an array called VIDEO%. This array is DIMensioned in line 245 to accommodate 80 columns by 22 rows. The scaled locations of each screen character is determined and stored in the array. To print the array, a simple PRINT command is issued from within a nested set of FOR-NEXT commands in line 1365-1367.

If a printer listing of the plot is desired, line 1375 is executed. This POKE disables the video screen and routes all output to the line printer. When finished, a POKE 3,149 in line 1375 restores the video screen to its normal operation.

If the (F)orecast option is selected in the action menu, the program is directed to line 1720 in line 1100. This routine is contained in lines 1720 through 1790. Both printer and monitor output are possible from this routine.

Line 1890 contains a sample DATA statement that may be used for the READ option for data input. This line may be removed if the R option is never utilized.

Lines 2000 through 2730 are used for screen promos and contain the various subroutines used in the data input menus.

# APPENDIX:
# Chapter 3—MENU

```
10 REM "MENU"
20 HOME:VTAB 2:HTAB 26:PRINT"APPLE CP/M ACCOUNTING SYSTEM":
   PRINT:PRINT:HTAB 34:PRINT"Version 83.1"
30 PRINT:PRINT:HTAB 29:PRINT"Steven Zimmerman, Ph.D.":
   PRINT:HTAB 34:PRINT"Leo M. Conrad":PRINT:HTAB 29:
   PRINT"Donald R. Smith, Ph. D., P.E.":VTAB 21:HTAB 23:
   INPUT "(Press < CR > to display  Main Menu)";DUM$
40 HOME:VTAB 2:HTAB 26:PRINT"APPLE CP/M ACCOUNTING SYSTEM":
   PRINT:HTAB 32:PRINT"*** MAIN MENU ***":PRINT:PRINT:
50 HTAB 8:PRINT"Select one  of the following:":PRINT:HTAB 21:
   PRINT"(C)heck writing Program":PRINT: HTAB 21:
   PRINT"(I)nvoice Program":PRINT:HTAB 21:
   PRINT"(M)ailing Label Program":PRINT:
60 HTAB 21:PRINT"(R)eports Income/Expense & Balance Sheet":
   PRINT:HTAB 21:PRINT"(S)et Up Chart of Accounts Program":
   PRINT:HTAB 21:PRINT"(T)ransaction Program":PRINT:PRINT:
70 HTAB 30:INPUT "(Enter C/I/M/R/S/T ) "; SL$
80 DD$="A"
90 IF LEFT$(SL$,1)="C" OR LEFT$(SL$,1)="c" THEN GOTO 210
100 IF LEFT$(SL$,1)="I" OR LEFT$(SL$,1)="i" THEN GOTO 160
110 IF LEFT$(SL$,1)="M" OR LEFT$(SL$,1)="m" THEN XX$=DD$+":
    MAIL.BAS":RUN XX$
120 IF LEFT$(SL$,1)="R" OR LEFT$(SL$,1)="r" THEN GOTO 160
130 IF LEFT$(SL$,1)="S" OR LEFT$(SL$,1)="s" THEN XX$=DD$+":
    SETUP.BAS":RUN XX$
140 IF LEFT$(SL$,1)="T" OR LEFT$(SL$,1)="t" THEN XX$=DD$+":
    TRANS.BAS":RUN XX$
150 PRINT CHR$(7):HTAB 10:PRINT"INVALID ENTRY...RE-ENTER":GOTO 70
160 HOME:PRINT CHR$(7):VTAB 5:
    PRINT"WARNING!...THIS PROGRAM REQUIRES 4 FILES TO BE OPEN"
170 PRINT:PRINT"YOU MUST RELOAD MBASIC WITH THE FOLLOWING COMMAND
    FROM CP/M":PRINT
180 PRINT"  FROM CP/M, TYPE THE FOLLOWING:"
185 IF LEFT$(SL$,1) = "I" OR LEFT$(SL$,1)="i" THEN PRINT:
    PRINT:PRINT"     MBASIC INVOICE/F:4":GOTO 200
190 PRINT:PRINT:PRINT"      MBASIC REPORT/F:4"
200 PRINT:PRINT:PRINT"To return to CP/M type SYSTEM <CR>":GOTO 300
210 HOME:PRINT CHR$(7):VTAB 5:
    PRINT"WARNING!...THIS PROGRAM REQUIRES 4 FILES TO BE OPEN"
220 PRINT:PRINT:PRINT"YOU MUST RELOAD MBASIC WITH THE FOLLOWING COMMAND
    FROM CP/M":PRINT
230 PRINT:PRINT:HTAB 10:PRINT"MBASIC CHECK/F:4":
    PRINT:PRINT:PRINT:PRINT"To return to CP/M type SYSTEM <CR>"
300 END

^@
```

# Chapter 4—SETUP

```
10 REM *** "CHART.BAS"...SETUP CHART OF ACCOUNTS PROGRAM ***
19 REM *** CLEAR ALL VARIABLES & STRINGS, DEFINE OUTPUT FORMATS ***
20 CLEAR:FMAT$="$$###,###,###,###.##"
29 REM *** DIMENSION TITLE OF ACCOUNTS ***
30 DIM TITLE(100):REM can have up to 100 title accounts
39 REM *** CLR SCREEN & PRINT FIRST SCREEN ***
40 HOME:VTAB 3:HTAB 26:
   PRINT"(S)etup Chart of Accounts Program":VTAB 8:HTAB 39
50 PRINT"by":VTAB 12:HTAB 27:PRINT"Steven M. Zimmerman, Ph.D.":
   PRINT:HTAB 33:PRINT"Leo M. Conrad":PRINT:HTAB 27:
   PRINT"Donald R. Smith, Ph.D., P.E."
60 VTAB 20:HTAB 25:INPUT "*** (PRESS < CR > TO BEGIN) ***";DUM$:HOME
69 REM *** BEGIN FILE NAME INPUT SECTION ***
70 VTAB 2:HTAB 33:PRINT"USER DIRECTIONS":PRINT:PRINT:
   HTAB 20:PRINT"1. Naming Files:":PRINT:HTAB 23:
   PRINT"Valid File Names-max. of 8 characters":
80 HTAB 23:PRINT"Followed by a period and 3 alpha characters":PRINT:
   PRINT"EXAMPLE:  FIRST.BAL"
90 PRINT:HTAB 20: PRINT"Valid Extent Codes are:":PRINT:HTAB 23:
   PRINT" .BAL  .INC    .DAT  .NUM  .INF":
99 REM *** INPUT FILE NAME AND VALIDATE IN GOSUB 1730
100 PRINT:PRINT:HTAB 20:
    INPUT "Bal. sheet or income/exp file name = ";F$:GOSUB 1730:
110 IF VC=0 GOTO 130 ELSE 120
120 PRINT CHR$(7):PRINT"*** INPUT FILE NAME ERROR...RE-ENTER":
    GOTO 100:
129 REM *** INPUT FILE NAME AND DISK A,B,C, OR D
130 :HTAB 20:PRINT"File name entered is "; F$
140 PRINT:PRINT:HTAB 20:INPUT "DISK  (A,B,C,D) ";DISK$:GOSUB 1780
150 IF VC=0 GOTO 170 ELSE 160
160 PRINT:PRINT:PRINT CHR$(7):
    PRINT"***** DISK DESIGNATOR ERROR enter (A/B/C/D)":GOTO 140
169 REM *** COMBINE FILE & EXTENT = FILE$ THE FILE NAME
170 FILE$=DISK$+":"+F$
179 REM *** DIMENSION THE VARIABLES -A# DENOTES DOUBLE PRECISION ***
180 PRINT:HTAB 5:INPUT "Maximum number of items in the Chart of
    Accounts";D:DIM A$(D),A#(D,2),A%(D)
189 REM *** INPUT FROM KEYBOARD OR DISK FILE ***
190 PRINT:PRINT:HTAB 5:
    INPUT "Select input from (D)isk or (K)eyboard "; FROM$
200 IF LEFT$(FROM$,1)="K" OR LEFT$(FROM$,1)="k" THEN GOSUB 260:
    GOTO 280
210 IF LEFT$(FROM$,1)<>"D" THEN IF LEFT$(FROM$,1)<> "d" THEN 190
219 REM ***READ CHART OF ACCTS FILE ***
220 Z=1:OPEN "I",1,FILE$:INPUT #1,DUMMY:REM input of file
230 INPUT #1,A%(Z),A$(Z),A#(Z,1),A#(Z,2)
240 IF LEFT$(A$(Z),3)="END" THEN Z=Z-1:CLOSE 1:GOTO 410
249 REM *** COUNT TOTAL NO. OF RECORDS IN THE FILE ***
250 Z=Z+1:GOTO 230:
```

```
259 REM ***SUB TO PRINT HEADINGS ***
260 PRINT:PRINT:PRINT"Entry No.";TAB(12);"Acct. No.";TAB(24);
    "Account Name";TAB(45);"Debit Amt.";TAB(63);"Credit Amt."
270 RETURN
279 REM *** INPUT DATA INTO CHART OF ACCOUNTS ***
280 Z=1
290 PRINT Z;:PRINT TAB(12);:INPUT ; A%(Z):IF A%(Z)<0
    THEN Z=Z-1:GOTO 410 ELSE PRINT TAB(23);:
300 INPUT ; A$(Z):PRINT TAB(44);:INPUT ;A#(Z,1):PRINT TAB(62);:
    INPUT A#(Z,2)
309 REM *** LIST THE CHART OF ACCOUNTS FILE ***
310 Z=Z+1:GOTO 290:REM Z is the total number of records in the file
320 PRINT:HTAB 5:PRINT"INPUT THE LINE NUMBERS TO BE LISTED":
    PRINT:HTAB 10:PRINT"STARTING LINE NO.";TAB(35);"ENDING LINE NO.":
330 PRINT TAB(15);:INPUT; A:PRINT TAB(40);:INPUT;B
340 PRINT CHR$(26):TAIL$="YES":GOSUB 260
349 REM *** LISTING ROUTINE FOR CHART OF ACCOUNTS ***
350 FOR I=A TO B
360 PRINT USING "####       ";I;
370 PRINT USING "###### ";A%(I);:
    PRINT TAB(24) USING "\            \";A$(I);:
380 PRINT TAB(39) USING "#,###,###,###.## ";A#(I,1);:
390 PRINT TAB(58) USING "#,###,###,###.## ";A#(I,2)
400 NEXT I:
409 REM *** SETUP SYSTEM MENU ***
410 PO=0:GOSUB 1820
420 IF LEFT$(CHOICE$,1)="C" OR LEFT$(CHOICE$,1)="c" THEN 580
430 IF LEFT$(CHOICE$,1)="L" OR LEFT$(CHOICE$,1)="l" THEN  320
440 IF LEFT$(CHOICE$,1)="A" OR LEFT$(CHOICE$,1)="a"
    THEN Z=Z+1:GOSUB 260:GOTO 290
450 IF LEFT$(CHOICE$,1)="N" OR LEFT$(CHOICE$,1)="n" THEN 1420
460 IF LEFT$(CHOICE$,1)="X" OR LEFT$(CHOICE$,1)="x" THEN 1560
470 IF LEFT$(CHOICE$,1)="R" OR LEFT$(CHOICE$,1)="r" THEN 620
480 IF LEFT$(CHOICE$,1)="T" OR LEFT$(CHOICE$,1)="t" THEN 1250
490 IF LEFT$(CHOICE$,1)="I" OR LEFT$(CHOICE$,1)="i" THEN 710
500 IF LEFT$(CHOICE$,1)="D" OR LEFT$(CHOICE$,1)="d" THEN 790
510 IF LEFT$(CHOICE$,1)="S" OR LEFT$(CHOICE$,1)="s" THEN PO=1:
    GOSUB 870
520 IF LEFT$(CHOICE$,1)="P" OR LEFT$(CHOICE$,1)="p" THEN 920
530 IF LEFT$(CHOICE$,1)="V" OR LEFT$(CHOICE$,1)="v" THEN 1160
540 IF LEFT$(CHOICE$,1)="E" OR LEFT$(CHOICE$,1)="e" THEN END
550 IF LEFT$(CHOICE$,1)="Z" OR LEFT$(CHOICE$,1)="z" THEN 1210
560 IF LEFT$(CHOICE$,1)="M" OR LEFT$(CHOICE$,1)="m" THEN 1120
570 GOTO 410
579 REM *** CHANGE OPTION SECTION ***
580 INPUT "INPUT  line number to be changed";L
590 GOSUB 260
600 PRINT L;:PRINT TAB(12);:INPUT;A%(L):PRINT TAB(21);:
    INPUT;A$(L):PRINT TAB(43);:INPUT ;A#(L,1):PRINT TAB(60);:
    INPUT A#(L,2)
610 A=L:B=L+1:GOTO 340
619 REM *** RECORD CURRENT FILE SECTION ***
620 OPEN"O",1,FILE$:PRINT #1,Z;
630 FOR I=1 TO Z
639 REM *** ROUND-OFF ROUTINE ***
```

```
640 FOR J=1 TO 2:X#=INT(100#*A#(I,J))/100#:DIF#=A#(I,J)-X#
650 IF DIF#>.005 THEN X#=X#+.01
660  A#(I,J)=X#:NEXT J
670 PRINT #1,A%(I);CHR$(34);A$(I);CHR$(34);A#(I,1);A#(I,2)
680  NEXT I
690 PRINT #1,0;CHR$(34);"END";CHR$(34);0;0
700 CLOSE 1:GOTO 410
709 REM *** INSERT ROUTINE ***
710 INPUT "Number of inserted line 0 to default";L:
    IF  L <1 OR L> Z THEN 410
720 GOSUB 260
730 Z=Z+1
740 FOR I=Z TO L+1 STEP -1:A%(I)=A%(I-1):A#(I,1)=A#(I-1,1):
    A#(I,2)=A#(I-1,2)
750 A$(I)=A$(I-1)
760 NEXT I
770 PRINT L;:PRINT TAB(15);:INPUT;A%(L):PRINT TAB(24);:
    INPUT;A$(L):PRINT TAB(40);:INPUT ;A#(L,1):PRINT TAB(54);:
    INPUT A#(L,2)
780 A=L-1:B=L+1:GOTO 340
789 REM *** DELETE RECORD ROUTINE ***
790 INPUT "Number of deleted line 0 to default";L:
    IF L<1 OR L>Z THEN 410
799 REM *** REDUCE RECORD COUNT BY ONE ***
800 Z=Z-1
810 FOR I=L TO Z:A%(I)=A%(I+1):A#(I,1)=A#(I+1,1):A#(I,2)=A#(I+1,2)
820 A$(I)=A$(I+1)
830 NEXT I
840 A%(Z+1)=0:A#(Z+1,1)=0:A#(Z+1,2)=0
850 A$(Z+1)="END"
860 A=L-1:B=L+1:GOTO 340
869 REM *** SUM DEBIT AND CREDITS ***
870 DEBIT#=0#:CREDIT#=0#:FOR I=1 TO Z:DEBIT#=DEBIT#+A#(I,1)
880 CREDIT#=CREDIT#+A#(I,2):NEXT I
890 IF PO=1 THEN PRINT "Sum of the debits: "; USING FMAT$; DEBIT#:
    PRINT "Sum of the credits: "; USING FMAT$; CREDIT#:
900 PRINT:PRINT "SUM OF THE CREDITS LESS SUM OF THE DEBITS:
    ";USING FMAT$;CREDIT#-DEBIT#
910 RETURN
919 REM *** LINE PRINTER ROUTINE  USING EPSON MX-80 OR 100 PRINTER ***
920 HOME:PRINT:PRINT:HTAB 15:
    INPUT "Align paper and READY printer...press RETURN when ready";DUM$
930 LPRINT: LPRINT :LPRINT:
940 INPUT "INPUT identification";ID$:LINE INPUT "Input date = ";DATE$:
950 LPRINT:LPRINT:LPRINT TAB(32) "PRINTOUT SUMMARY":
    LPRINT TAB(32) "FILE NAME: ";FILE$
960 LPRINT:LPRINT "Identification: ";ID$; TAB(60);"DATE: ";DATE$:
    LPRINT:
970 LPRINT TAB(4) "No.";TAB(9);"Account No.";TAB(22);"Account Name";
    TAB(48);"Debit Amount";TAB(67);"Credit Amount":
980 DEBIT#=0#:CREDIT#=0#:BAL#=0#
990 FOR I=1 TO Z
1000 LPRINT USING "#####";I;:LPRINT TAB(11) USING "###### ";A%(I);:
1010 LPRINT TAB(21) USING "\                    \";A$(I);:
1020 LPRINT TAB(44) USING "#,###,###,###.## ";A#(I,1);:
```

```
1030 LPRINT TAB(63) USING "#,###,###,###.## ";A#(I,2)
1040 DEBIT#=DEBIT#+A#(I,1):CREDIT#=CREDIT#+A#(I,2)
1050 NEXT I
1060 LPRINT TAB(44) "-----------------  -----------------"
1070 LPRINT TAB(44) USING "$$#,###,###,###.## ";DEBIT#,CREDIT#
1080 LPRINT TAB(44)"=================  ================="
1090 LPRINT :LPRINT "Sum of the Credits less sum of the Debits: ";
     USING FMAT$;CREDIT#-DEBIT#
1100 LPRINT CHR$(12):REM FORMS FEED FOR EPSON MX-80 & 100 PRINTER
1110 GOTO 410
1111 REM *** END OF LINE PRINTER ROUTINE ***
1119 REM *** MERGING SECOND FILE WITH CURRENT FILE ROUTINE ***
1120 INPUT "File to be added ";F$:INPUT "Disk (A,B,C,D)";DXXX$
1130 AFILE$=DXXX$+":"+F$
1140 OPEN "I",1,AFILE$:INPUT #1,DUMMY
1150 Z=Z+1:GOTO 230
1159 REM *** CHANGING RECORD INFORMATION SECTION ***
1160 INPUT "INPUT  line number to be changed";L
1170 GOSUB 260
1180 PRINT L;:PRINT TAB(12);:PRINT USING "######";A%(L);:
     PRINT TAB(24);:PRINT A$(L);:PRINT TAB(44);:
1190 INPUT ;A#(L,1): PRINT TAB(62);:INPUT A#(L,2)
1200 A=L-1:B=L+1:GOTO 340
1209 REM *** ZERO VALUES SECTION ***
1210 PRINT:PRINT"This routine is for year ending zeroing all
     income/expenses to date information":PRINT :
1220 INPUT "YES IF YOU ARE SURE YOU WISH TO CONTINUE";YY$:
     IF YY$<>"YES" THEN 340
1230 FOR I=1 TO Z:A#(I,1)=0#:A#(I,2)=0#:NEXT I
1240 WIDTH(80):GOTO 410
1249 REM *** TITLE ACCOUNTS SECTION ***
1250 PRINT :PRINT :INPUT " (N)ew file or (U)pdate ";NNN$
1260 NZ$=DISK$+":DATAF.INF"
1270 IF LEFT$(NNN$,1)="N" OR LEFT$(NNN$,1)="n" THEN 1310
1280 OPEN"I",1,NZ$
1290 INPUT #1,NOTITLE:FOR I=1 TO NOTITLE:INPUT #1,TITLE%(I):
     NEXT I:CLOSE 1
1300 GOTO 1330
1310 PRINT :PRINT :INPUT "Number of title accounts";NOTITLE
1320 FOR I=1 TO NOTITLE:PRINT "Input title account number ";I;" .";:
     INPUT TITLE%(I):NEXT I
1330 PRINT :PRINT "Number","Title number"
1340 PRINT :FOR I=1 TO NOTITLE:PRINT I,TITLE%(I):NEXT I
1350 INPUT "99 to continue or number to correct";KK:IF KK=99 THEN 1380
1360 INPUT "New title number";TITLE%(KK)
1370 GOTO 1330
1380 PRINT "Information being saved in file named ";NZ$
1390 OPEN"O",1,NZ$
1400 PRINT #1,NOTITLE:FOR I=1 TO NOTITLE:PRINT #1,TITLE%(I):
     NEXT I:CLOSE 1
1410 GOTO 410
1419 REM *** NUMBERING ACCOUNTS SECTION ***
1420 PRINT :PRINT :
     PRINT "Input the highest account number for each of the":
     PRINT" following types of accounts"
```

```
1430 INPUT "(N)ew series or (U)pdate ";QQ$
1440 IF LEFT$(QQ$,1)="U" OR LEFT$(QQ$,1)="u" THEN X$=DISK$+":
     ACCT.NUM":OPEN "I",1,X$:FOR I=0 TO 4:INPUT #1,HNO(I):
     NEXT I:CLOSE 1
1450 AX$(0)="Assets":AX$(1)="Liabilities":AX$(2)="S. H. EQUITY"
1460 AX$(3)="Income":AX$(4)="Expenses"
1470 IF LEFT$(QQ$,1)="U" OR LEFT$(QQ$,1)="u" THEN 1490
1480 FOR I=0 TO 4:PRINT AX$(I);:INPUT HNO(I):NEXT I
1490 PRINT "Number","Type",,"Highest Account Number"
1500 FOR I=0 TO 4:PRINT I,AX$(I),HNO(I):NEXT I
1510 INPUT "99 to continue or number to correct";KK:
     IF  KK=99 THEN 1530
1520 PRINT AX$(KK);:INPUT HNO(KK):GOTO 1490
1530 NZ$=DISK$+":ACCT.NUM":OPEN "O",1,NZ$
1540 FOR I=0 TO 4:PRINT #1,HNO(I):NEXT I:CLOSE 1
1550 GOTO 410
1559 REM *** COMPANY NAME AND ADDRESS SECTION ***
1560 PRINT :INPUT " (N)ew name or (U)pdate ";UX$:X$=DISK$+":NAMEX.DAT"
1570 IF LEFT$(UX$,1)="N" OR LEFT$(UX$,1)="n" THEN 1600
1580 OPEN "I",1,X$:FOR I=0 TO 5:INPUT #1,COMPANY$(I):NEXT I:CLOSE 1
1590 GOTO 1660
1600 LINE INPUT "Company's Name:";COMPANY$(0)
1610 LINE INPUT "CO. Address: ";COMPANY$(1)
1620 LINE INPUT "City: ";COMPANY$(2)
1630 LINE INPUT "State: ";COMPANY$(3)
1640 LINE INPUT "Zip Code: ";COMPANY$(4)
1650 LINE INPUT "Phone number";COMPANY$(5)
1660 PRINT "Number","Data"
1670 FOR I=0 TO 5:PRINT I,COMPANY$(I):NEXT I
1680 INPUT "99 to continue or number to change";XX:IF  XX=99 THEN 1700
1690 INPUT "New data";COMPANY$(XX):GOTO 1660
1700 OPEN "O",1,X$
1710 FOR I=0 TO 5:PRINT#1,CHR$(34);COMPANY$(I);CHR$(34);:NEXT I:CLOSE 1
1720 GOTO 410:COMPANY$(XX):GOTO 1660
1760 VC=1:RETURN
1770 VC=0:RETURN
1780 REM VAL DISK$ ENTRY FOR EITHER A,B,C,OR D
1790 IF DISK$="A" OR DISK$="B" OR DISK$="C" OR DISK$="D"
     GOTO 1810 ELSE 1800
1800 VC=1:RETURN
1810 VC=0:RETURN
1820 REM  *** SYSTEM MENU EDIT SUBROUTINE ***
1830 PRINT"SELECT...(A)dd,(C)hange,(D)elete,(E)nd,(I)nsert,(L)ist,
     (M)erge,(N)o Asset/Liab":
1840 HTAB 17:PRINT"(P)rint,(R)ecord,(S)um debits/credits,
     (T)itle Accounts,":
1850 HTAB 19:PRINT"(V)alues change,(X)company's name,(Z)ero all values"
1860 HTAB 30:INPUT "ENTER your choice ";CHOICE$
1870 RETURN
1880 END

^@
```

# Chapter 5—TRANS

```
10 REM "TRANS"
20 A$="### ####  \       \ \          \ ####  ####  ###,###,
   ###.##   ###,###,###.## \\":AFORM$="\    \"
30 UFORMAT$="#,###,###.##"
40 XFORM$="\    \"
50 AC$="### #### \  \ \    \ #### #### #,###,###.##  #,
   ###,###.## \\"
60 GOSUB 1670
70 PRINT:HTAB 5:INPUT "Transaction file name: ";F$:PRINT:
80 HTAB 5:INPUT "Disk (A,B,C,D)";DISK$:GOSUB 1630:PRINT:
90 IF VC=0 GOTO 110 ELSE 100
100 PRINT CHR$(7):
    PRINT"***** DISK DESIGNATOR ERROR enter (A/B/C/D) ":GOTO 80
110 FILE$=DISK$+":"+F$
120 HTAB 5:LINE INPUT "DATE  ?";DATE$:PRINT:
130 HTAB 5:INPUT "Maximum number of transactions";D:
    DIM A%(D,3),A$(D,3),A#(D,2)
140 PRINT :HTAB 5:INPUT "Select input from (D)isk or (K)eys ";FROM$
150 IF LEFT$(FROM$,1)="K" OR LEFT$(FROM$,1)="k" THEN GOSUB 210:
    GOTO 270
160 IF LEFT$(FROM$,1)<>"D" THEN IF LEFT$(FROM$,1)<>"d" THEN 140
170 Z=1:OPEN "I",1,FILE$
180 INPUT #1,A%(Z,1),A$(Z,1),A$(Z,2),A%(Z,2),A%(Z,3),A#(Z,1),
    A#(Z,2),A$$3)
190 IF LEFT$(A$(Z,1),3)="END" THEN Z=Z-1:CLOSE:GOTO 370
200 Z=Z+1:GOTO 180:REM Z is the total number of records in the file
210 IF TAIL$="YES" THEN PRINT "No  ";
220 PRINT"Trans Date";TAB(16);"Description";TAB(32);"Acct# Acct#";
    TAB(46);"Amount Db.";TAB(60);"Amount Cr.";:
    IF TAIL$="YES" THEN PRINT TAB(74);"STATUS" ELSE PRINT
230 PRINT" No.";TAB(33);"Db.   Cr."
240 PRINT"---- -----  ------------------ ----- ----- --------------
    --------------":PRINT:
250 IF TAIL$="YES" THEN PRINT "    ";
260 PRINT "No":RETURN
270 Z=1
280 INPUT;A%(Z,1):IF A%(Z,1)<1 THEN PRINT:PRINT:Z=Z-1:
    GOTO 370 ELSE PRINT TAB(6);:INPUT;A$(Z,1):INPUT;A$(Z,2):
    PRINT TAB(32);:
290 INPUT;A%(Z,2):PRINT TAB(38);:INPUT;A%(Z,3):
    PRINT TAB(44);:INPUT ;A#(Z,1):PRINT TAB(59);:
    INPUT A#(Z,2):A$(Z,3)="ISSUED"
300 Z=Z+1:GOTO 280:REM Z is the total number of records in the file
310 PRINT:HTAB 5:PRINT"INPUT THE LINE NUMBERS TO BE LISTED":PRINT:
320 HTAB 10:PRINT"STARTING LINE NUMBER";TAB(35);"ENDING LINE NUMBER":
330 PRINT TAB(15);:INPUT;A:PRINT TAB(40);:INPUT B:PRINT:PRINT:
340 TAIL$="YES":GOSUB 210:FOR I=A TO B
350 PRINT USING A$;I,A%(I,1),A$(I,1),A$(I,2),A%(I,2),A%(I,3),A#(I,1),
    A##,2),A$(I,3)
```

```
360 NEXT I
370 PO=0:PRINT:PRINT:PRINT"(A)dd,(B)alance,(C)hange,(D)elete,(E)nd,
    (I)nsert, (K)lear,(L)ist,(M)erge,(P)rint,":
380 HTAB 20:PRINT"(R)ecord,(S)um debits/credits, (V)alues":PRINT:
390 HTAB 24:INPUT "ENTER YOUR CHOICE ";CHOICE$:PRINT:PRINT:
400 IF LEFT$(CHOICE$,1)="C" OR LEFT$(CHOICE$,1)="c" THEN 540
410 IF LEFT$(CHOICE$,1)="K" OR LEFT$(CHOICE$,1)="k" THEN 1230
420 IF LEFT$(CHOICE$,1)="V" OR LEFT$(CHOICE$,1)="v" THEN 1190
430 IF LEFT$(CHOICE$,1)="L" OR LEFT$(CHOICE$,1)="l" THEN 310
440 IF LEFT$(CHOICE$,1)="B" OR LEFT$(CHOICE$,1)="b" THEN 1340
450 IF LEFT$(CHOICE$,1)="A" OR LEFT$(CHOICE$,1)="a" THEN Z=Z+1:
    TAIL$="NO":GOSUB 210:GOTO 280
460 IF LEFT$(CHOICE$,1)="R" OR LEFT$(CHOICE$,1)="r" THEN 580
470 IF LEFT$(CHOICE$,1)="I" OR LEFT$(CHOICE$,1)="i" THEN 670
480 IF LEFT$(CHOICE$,1)="D" OR LEFT$(CHOICE$,1)="d" THEN 760
490 IF LEFT$(CHOICE$,1)="S" OR LEFT$(CHOICE$,1)="s" THEN PO=1:
    GOSUB 840
500 IF LEFT$(CHOICE$,1)="P" OR LEFT$(CHOICE$,1)="p" THEN 880
510 IF LEFT$(CHOICE$,1)="E" OR LEFT$(CHOICE$,1)="e" THEN END
520 IF LEFT$(CHOICE$,1)="M" OR LEFT$(CHOICE$,1)="m" THEN 1150
530 GOTO 370
540 INPUT "INPUT  line number to be changed";L
550 TAIL$="YES":GOSUB 210
560 INPUT;A%(L,1):INPUT;A$(L,1):INPUT ;A$(L,2):PRINT TAB(34);:
    INPUT ;A%(L,2):PRINT TAB(43);:INPUT ;A%(L,3):
570 A=L-1:B=L+1:GOTO 340
580 OPEN"O",1,FILE$
590 FOR I=1 TO Z
600 FOR J=1 TO 2:X#=INT(100#*A#(I,J))/100#:DIF#=A#(I,J)-X#
610 IF DIF#>5.000000000000005D-03 THEN X#=X#+.01#
620 A#(I,J)=X#:NEXT J
630 PRINT #1,A%(I,1);CHR$(34);A$(I,1);CHR$(34);CHR$(34);A$(I,2);
    CHR$(34)A%(I,2);A%(I,3);A#(I,1);A#(I,2);CHR$(34);A$(I,3);
    CHR$(34);
640 NEXT I
650 PRINT #1,0;CHR$(34);"END";CHR$(34);CHR$(34);"END"CHR$(34);
    0;0;0;0;CHR$(34);"END";CHR$(34)
660 CLOSE:GOTO 370
670 INPUT "Number of inserted line 0 to default";L:
    IF  L <1 OR L> Z THEN 370
680 Z=Z+1
690 FOR I=Z TO L+1 STEP -1:A#(I,1)=A#(I-1,1):A#(I,2)=A#(I-1,2)
700 A%(I,1)=A%(I-1,1):A%(I,2)=A%(I-1,2):A%(I,3)=A%(I-1,3)
710 A$(I,1)=A$(I-1,1):A$(I,2)=A$(I-1,2):A$(I,3)=A$(I-1,3)
720 NEXT I
730 TAIL$="YES":GOSUB 210
740 INPUT;A%(L,1):INPUT;A$(L,1):INPUT ;A$(L,2):PRINT TAB(34);:
    INPUT ;A%(L,2):PRINT TAB(43);:INPUT ;A%(L,3):PRINT TAB(56);:
750 A=L-1:B=L+1:GOTO 340
760 INPUT "Number of deleted line 0 to default";L:
    IF L <1 OR L>Z THEN 370
770 Z=Z-1
780 FOR I=L TO Z:A#(I,1)=A#(I+1,1):A#(I,2)=A#(I+1,2):
    A%(I,1)=A%(I+1,1):A%(I,2)=A%(I+1,2):A%(I,3)=A%(I+1,3):
    :A$(I,1)=A$(I+1,1):
```

```
790 NEXT I
800 A#(Z+1,1)=0:A#(Z+1,2)=0
810 A%(Z+1,1)=0:A%(Z+1,2)=0:A%(Z+1,3)=0
820 A$(Z+1,1)="END":A$(Z+1,2)="END":A$(Z+1,3)="END"
830 A=L-1:B=L+1:GOTO 340
840 DEBIT#=0#:CREDIT#=0#:FOR I=1 TO Z:DEBIT#=DEBIT#+A#(I,1)
850 CREDIT#=CREDIT#+A#(I,2):NEXT I
860 IF PO=1 THEN PRINT "Sum of the debits: ";DEBIT#:
    PRINT "Sum of the credits: ";CREDIT#
870 RETURN
880 HTAB 5:INPUT "INPUT Identification of run";ID$:LPRINT:
890 LPRINT "**** IDENTIFICATION:   ";ID$
900 LPRINT:LPRINT "Date:  ";DATE$;:LPRINT TAB(65);"File:  ";FILE$:
    LPRINT
910 LPRINT :LPRINT :LPRINT :
920 HTAB 5:PRINT "Printer menu":HTAB 5:
    PRINT "     (A)ll transactions":HTAB 5:
    INPUT "     (O)ne account";AX$:PRINT :PRINT :
930 HTAB 5:INPUT "   CASH Account No is ";CASH%:HTAB 8:
    INPUT "Amount of ORIGINAL Balance: ";BAL#
940 LPRINT:LPRINT "Original Balance: ";USING UFORMAT$;BAL#
950 IF LEFT$(AX$,1)="O" OR LEFT$(AX$,1)="o" THEN
    INPUT "Account number",ACCNO
960 GOSUB 970:GOTO 1010
970 LPRINT :LPRINT "Trans   Date  DESCRIP ACCOUNT","AMOUNT";:LPRINT "
980 LPRINT "      ";
990 LPRINT "No","       DEBIT CREDIT    DEBIT        CREDIT":
    DEBIT#=0#:CREDIT#=0#
1000 RETURN
1010 FOR I=1 TO Z
1020 IF A%(I,1)=ACCNO OR LEFT$(AX$,1)="A" OR LEFT$(AX$,1)="a" THEN
     LPRINT USING AC$;I,A%(I,1),A$(I,1),A$(I,2),A%(I,2),A%(I,3),
     A#(I,1),A#(I,2),A$(I,3);:DEBIT#=DEBIT#+A#(I,1):CREDIT#=CREDIT#
     +A#(I,2) ELSE 1070
1040 IF CASH%=A%(I,2) THEN BAL#=BAL#+A#(I,1)
1050 IF CASH%=A%(I,3) THEN BAL#=BAL#-A#(I,2)
1060 LPRINT USING "  ###,###,###.##";BAL#
1070 NEXT I
1080 GOSUB 1090:GOTO 370
1090 LPRINT TAB(30)"--------------      -------------"
1100 LPRINT TAB(30) USING "###,###,###.##  ";DEBIT#,CREDIT#
1110 LPRINT TAB(30)"==============      ============="
1120 REM  CHR$(12) IS A FORMS FEED FOR THE EPSON PRINTER
1130 LPRINT CHR$(12)
1140 RETURN
1150 HTAB 5:INPUT "File to be added ";F$:PRINT:HTAB 5:
     INPUT "Disk (A,B,C,D)";DISK$
1160 AFILE$=DISK$+":"+F$
1170 OPEN "I",1,AFILE$
1180 Z=Z+1:GOTO 180
1190 INPUT "INPUT  line number to be changed";L
1200 TAIL$="YES":GOSUB 210
1210 PRINT A%(L,1);:PRINT A$(L,1);:PRINT A$(L,2);:PRINT TAB(34);:
     INPUT ;A%(L,2):PRINT TAB(43);:INPUT ;A%(L,3):PRINT TAB(56);:
1220 A=L:B=L+1:GOTO 340
```

```
1230 TAIL$="YES":INPUT "Input transaction starting Number";L:
     L=L-1:GOSUB 210
1240 L=L+1:PRINT
1250 PRINT A%(L,1);:PRINT A$(L,1);" ";:PRINT A$(L,2);:
     PRINT TAB(34);:PRINT A%(L,2);:PRINT TAB(43);:PRINT A%(L,3);:
1260 PRINT :INPUT "(A)djustment,(C)leared or (I)ssued,(E)nd ";KK$
1270 IF LEFT$(KK$,1)="C" OR LEFT$(KK$,1)="c" THEN A$(L,3)="CLEAR":
     GOTO 1320
1280 IF LEFT$(KK$,1)="a" OR LEFT$(KK$,1)="A" THEN A$(L,3)="ADJUST":
     GOTO 1320
1290 IF LEFT$(KK$,1)="I" OR LEFT$(KK$,1)="i" THEN A$(L,3)="ISSUED":
     GOTO 1320
1300 IF LEFT$(KK$,1)="e" OR LEFT$(KK$,1)="E" THEN GOTO 370
1310 GOTO 1260
1320 IF L<Z THEN 1240
1330 GOTO 370
1340 HOME:PRINT:PRINT:
1341 PRINT"The objective of this routine is to balance the values
     of your cash acct.":PRINT:
1350 PRINT
1360 PRINT"It is assumed that you have created a future action file,
     have recorded"
1361 PRINT"your current period's file, identified the cleared
     transactions in your"
1362 PRINT"current period's file, and have an old, uncleared file.":
1370 PRINT :PRINT
1380 INPUT "     (D)elete cleared transactions,(P)rint results ";KK$
1390 IF LEFT$(KK$,1)="P" OR LEFT$(KK$,1)="p" THEN 1480
1400 IF LEFT$(KK$,1)="D" OR LEFT$(KK$,1)="d" THEN 1420
1410 GOTO 1380
1420 PRINT :INPUT "Name of file with uncleared transactions";X$:
     X$=DISK$+":"+X$
1430 OPEN "O",1,X$
1440 PRINT "File of uncleared transaction is called:  ";X$
1442 FOR I=1 TO Z:IF LEFT$(A$(I,3),1)="I" OR LEFT$(A$(I,3),1)="i"
     THEN PRINT #1,A%(I,1);CHR$(34);A$(I,1);CHR$(34);CHR$(34);
     A$(I,2);CHR$(34);A%(I,2);A%(I,3);A#(I,1);A#(I,2);CHR$(34);
     A$(I,3);CHR$(34);
1460 NEXT I:PRINT #1,0;CHR$(34);"END";CHR$(34);CHR$(34);"END";
     CHR$(34);0;0;0;CHR$(34);"ENN;CHR$(34):CLOSE 1
1470 GOTO 370
1480 INPUT "INPUT Identification of run";ID$:LPRINT :LPRINT ID$:
     INPUT "Number of cash account";CASH%
1490 INPUT "Ending balance on balance sheet";BAL#:
     LPRINT "Ending balance on balance sheet ";
     USING"###,###,###,###.##";BAL#
1500 LPRINT :LPRINT "Cash account reconciliation account # ";CASH%:
     LPRINT :LPRINT DATE$:LPRINT :LPRINT:LPRINT "File: ";FILE$:
1510 GOSUB 970
1520 FOR I=1 TO Z
1530 LPRINT USING AC$;I,A%(I,1),A$(I,1),A$(I,2),A%(I,2),A%(I,3),
     A#(I,1))#(I,2),A$(I,3);:DEBIT#=DEBIT#+A#(I,1):
1540 IF CASH%=A%(I,3) THEN IF LEFT$(A$(I,3),1)="i" OR
     LEFT$(A$(I,3),1)="I" THEN BAL#=BAL#+A#(I,1)
```

```
1550 IF CASH%=A%(I,2) THEN IF LEFT$(A$(I,3),1)="i" OR
     LEFT$(A$(I,3),1)="I" THEN BAL#=BAL#-A#(I,1)
1560 IF CASH%=A%(I,2) THEN IF LEFT$(A$(I,3),1)="a" OR
     LEFT$(A$(I,3),1)="A" THEN BAL#=BAL#+A#(I,1)
1570 IF CASH%=A%(I,3) THEN IF LEFT$(A$(I,3),1)="a" OR
     LEFT$(A$(I,3),1)="A" THEN BAL#=BAL#-A#(I,1)
1580 LPRINT USING "  ###,###,###.##";BAL#
1590 NEXT I
1600 GOSUB 1090
1610 LPRINT :LPRINT "Difference bank's balance and adjusted is ";
     BANK#;" - ";BAL#;" = ";USING "###,###,###.##";BANK#-BAL#
1620 GOTO 370
1630 REM VALIDATE DISK ENTRY FOR A,B,C, OR D
1640 IF DISK$="A" OR DISK$="B" OR DISK$="C" OR DISK$="D"
     GOTO 1660 ELSE 1650
1650 VC=1:RETURN
1660 VC=0:RETURN
1670 REM FIRST SCREEN
1680 HOME:VTAB 3:HTAB 30:PRINT"TRANSACTION ROUTINE":VTAB 8:HTAB 39:
1690 PRINT"by": VTAB 12:HTAB 27:PRINT"Steven M. Zimmerman, Ph.D."
1700 PRINT:HTAB 33:PRINT"Leo M. Conrad"
1710 PRINT:HTAB 27:PRINT"Donald R. Smith, Ph.D., P.E.":
1720 VTAB 20:HTAB 25:INPUT "*** (press RETURN to begin) ***";DUM$:HOME:
1730 RETURN

^@
```

# Chapter 6—REPORT

```
10 REM "REPORT generation program"
20 HOME:CLEAR 2000:WIDTH 80:GOSUB 1530
30 REM GOSUB TO INSTRUCTIONS
40 GOSUB 1610
50 HTAB 5:INPUT "1. The Current Date (xx/xx/xx) is ";DATE$:PRINT:
60 MAT$="###,###,###.## ":FMAU$="\       \"
70 FORM$=\                  \"
80 MAX$=" $$###,###,###.## "
90 HTAB 5:PRINT"2. Name of current TRANSACTION file with extender is:
   ";:INPUT TRF$
100 PRINT :HTAB 5:PRINT "3. Name of old BALANCE sheet file
    without extender";:INPUT OLDBALF$
110 PRINT :HTAB 5:PRINT "4. Name of new BALANCE sheet file
    without extender";:INPUT NEBALF$
120 PRINT :HTAB 5:INPUT"5. DISK (A,B,C,D)";DD$:PRINT:HTAB 5:
    PRINT"***** Opening the Files ***** ":
130 TRF$=DD$+":"+TRF$
140 OLDPAL$=DD$+":"+OLDBALF$+".INC"
150 OLDBALF$=DD$+":"+OLDBALF$+".BAL"
160 NEPAL$=DD$+":"+NEBALF$+".INC"
170 NEBALF$=DD$+":"+NEBALF$+".BAL"
180 PRINT :PRINT :PRINT "The Balance sheet file  extender = .BAL"
190 PRINT :PRINT "The Income and expense  extender = .INC "
200 OPEN "I",1,OLDBALF$:INPUT #1,NOBAL
210 OPEN "I",2,OLDPAL$:INPUT #2,NOPAL:TTNO=NOBAL+NOPAL:
    DIM A%(TTNO),A#(TTNO,2),A$(TTNO),B#(TTNO,2):
220 REM A$ is information todate, B# is current period information
230 OPEN "I",3,TRF$
240 FOR I=1 TO NOBAL:INPUT #1,A%(I),A$(I),A#(I,1),A#(I,2):
    NEXT I:CLOSE 1
250 FOR I=NOBAL+1 TO TTNO:INPUT #2,A%(I),A$(I),A#(I,1),A#(I,2):
    NEXT I:CLOSE 2
260 INPUT #3,AA%(1),AA$,AA$,AA%(2),AA%(3),AA#(1),AA#(2),XX$:
    IF AA$="END" THEN CLOSE 3:GOTO 350
270 A1=0:A2=0:I=0:REM Posting routine
280 I=I+1:IF AA%(2)=A%(I) THEN B#(I,1)=B#(I,1)+AA#(1):A1=1
290 IF AA%(3)=A%(I) THEN B#(I,2)=B#(I,2)+AA#(2):A2=1
300 IF A1+A2=2 THEN 260
310 IF I<TTNO THEN 280
320 PRINT "There is no match for transaction #";I
330 PRINT AA%(1),AA$,AA%(2),AA%(3),AA#(1),AA#(2),XX$
340 CLOSE :END
350 REM All transactions have been input and posted
360 FOR I=1 TO TTNO:IF B#(I,1)>=B#(I,2) THEN B#(I,1)=B#(I,1)-B#(I,2):
    B#(I,2)=0#
370 IF B#(I,1)<B#(I,2) THEN B#(I,2)=B#(I,2)-B#(I,1):B#(I,1)=0#
380 NEXT I
390 NZ$=DD$+":ACCT.NUM"
400 OPEN "I",1,NZ$
```

```
410 FOR I=1 TO 5:INPUT #1,ACC%(I):NEXT I:CLOSE 1
420 REM ACC%(i)-- maximum account numbers
430 REM all current period are now organized must now update
    year to date file
440 FOR I=1 TO NOBAL+NOPAL
450 A#(I,2)=A#(I,2)+B#(I,2):A#(I,1)=A#(I,1)+B#(I,1)
460 NEXT I
470 FOR I=1 TO TTNO:IF A#(I,1)>=A#(I,2) THEN
    A#(I,1)=A#(I,1)-A#(I,2):A#(I,2)=0#
480 IF A#(I,1)<A#(I,2) THEN A#(I,2)=A#(I,2)-A#(I,1):A#(I,1)=0#
490 NEXT I
500 GOSUB 1710
510 LPRINT TAB(39)DATE$
520 LPRINT
530 NZ$=DD$+":NAMEX.DAT"
540 OPEN "I",1,NZ$
550 FOR I=0 TO 5:INPUT #1,COMPANY$(I):NEXT  I:CLOSE 1:
    GOSUB 560:GOTO 590
560 LPRINT TAB(35) COMPANY$(0)
570 LPRINT TAB(35) COMPANY$(1)
580 LPRINT TAB(35) COMPANY$(2);",";COMPANY$(3);" ";COMPANY$(4):
    LPRINT:RETURN
590 LPRINT TAB(20)"I n c o m e     &    E x p e n s e
        S t a t e m e n t"
600 LPRINT :LPRINT ,,,"I n c o m e  "
610 NZ$=DD$+":DATAF.INF":OPEN "I",1,NZ$:
    REM to input the title accounts
620 INPUT #1,NOTITLE:DIM TITLE%(NOTITLE+1):FOR I=1 TO NOTITLE:
    INPUT #1,TITLE%(I):NEXT I:
    CLOSE 1
630 LPRINT :LPRINT :LPRINT
640 LPRINT "Number","Account","Current Period","Year to date"
650 LPRINT ,"          Amount","Percent"," Amount","   Percent"
660 J=0
670 J=J+1:IF TITLE%(J)<ACC%(3) AND J < NOTITLE THEN 670
680 II=NOBAL:CREDIT#=0#:INCOME#=0#
690 II=II+1
700  CREDIT#=CREDIT#+B#(II,2):INCOME#=INCOME#+A#(II,2)
710 IF ACC%(4)<>A%(II) THEN 690
720 I=NOBAL
730 I=I+1
740 IF I=NOBAL+NOPAL THEN PRINT "Trouble should not have ended":END
750 IF A%(I)=TITLE%(J) THEN J=J+1:LPRINT :LPRINT A$(I):
    LPRINT :GOTO 730
760 IF CREDIT#>0 THEN PERCENX=B#(I,2)/CREDIT# ELSE PERCENX =0
770 IF INCOME#>0 THEN PERCENT=A#(I,2)/INCOME# ELSE PERCENT =0
780 LPRINT A%(I);:LPRINT USING FMAU$;A$(I);:
    LPRINT USING MAT$;B#(I,2),PERCENX,A#(I,2),PERCENT
790 IF ACC%(4)<>A%(I) THEN 730
800 LPRINT ,"       ----------","         ------------"
810 LPRINT ,USING MAX$;CREDIT#;:
    LPRINT ;"              ";USING MAX$;INCOME#
820 LPRINT ,"       ==========","         ============"
830 LPRINT :LPRINT
840 LPRINT ,,,"E x p e n s e s "
```

```
850 LPRINT " ":LPRINT " ":LPRINT "Number","Account","Current Period",
   "Year to date"
860 LPRINT ,"          Amount","Percent"," Amount","   Percent":
    DEBIT#=0#:EXPENSE#=0#
870 II=II+1:IF II>NOBAL+NOPAL THEN GOTO 900
880 DEBIT#=DEBIT#+B#(II,1):EXPENSE#=EXPENSE#+A#(II,1)
890 IF ACC%(4)<>A%(II) THEN 870
900 I=I+1:IF I>NOBAL+NOPAL THEN GOTO 960
910 IF A%(I)=TITLE%(J) THEN J=J+1:LPRINT :LPRINT A$(I):
    LPRINT :GOTO 900
920 IF DEBIT#>0 THEN PERCENX=B#(I,1)/DEBIT# ELSE PERCENX =0
930 IF EXPENSE#>0 THEN PERCENT=A#(I,1)/EXPENSE# ELSE PERCENT =0
940 LPRINT A%(I);:LPRINT USING FMAU$;A$(I);:
    LPRINT USING MAT$;B#(I,1),PERCENX,A#(I,1),PERCENT
950 IF ACC%(4)<>A%(I) THEN 900
960 LPRINT ,"      ----------","         ------------"
970 LPRINT ,USING MAX$;DEBIT#;:LPRINT;"              ";USING MAX$;
    EXPENSE#
980 LPRINT ,"      ==========","         ============"
990 LPRINT :LPRINT
1000 PER#=CREDIT#-DEBIT#
1010 PROFIT#=INCOME#-EXPENSE#
1020 LPRINT "Profit              ",USING MAX$;PER#;
1030 LPRINT;"              ";USING MAX$;PROFIT#
1040 REM closing profit/loss into stockholder's equity
1050 X=NOBAL:A#(X,2)=A#(X,2)+PER#
1060 IF A#(X,1)>A#(X,2) THEN A#(X,1)=A#(X,1)-A#(X,2):A#(X,2)=0
1070 IF A#(X,1)<=A#(X,2) THEN A#(X,2)=A#(X,2)-A#(X,1):A#(X,1)=0
1080 LPRINT :LPRINT :LPRINT :LPRINT
1090 LPRINT TAB(35)DATE$:GOSUB 560
1100 LPRINT :LPRINT ,,"B a l a n c e     S h e e t   "
1110 LPRINT :LPRINT
1120 LPRINT :LPRINT :LPRINT "Number","Account"
1130 LPRINT ,,,"      Debit","Credit"
1140 I=0:J=1:DEBIT#=0#:CREDIT#=0#
1150 I=I+1:IF I>NOBAL THEN 1200
1160 IF A%(I)=TITLE%(J) THEN J=J+1::LPRINT :LPRINT A$(I):
     LPRINT :GOTO 1150
1170 LPRINT A%(I),USING FORM$;A$(I);:LPRINT USING MAT$;A#(I,1);:
     LPRINT USING MAT$;A#(I,2):DEBIT#=DEBIT#+A#(I,1):
1180 CREDIT#=CREDIT#+A#(I,2)
1190 IF ACC%(4)<>A%(I) THEN 1150
1200 LPRINT ,,,"----------"," ------------"
1210 LPRINT ,,"          " ;USING MAT$;DEBIT#;:LPRINT USING MAT$;CREDIT#
1220 LPRINT ,,,"==========","  ============"
1230 LPRINT :LPRINT
1240 LPRINT "Computer files used are"
1250 LPRINT "     Transaction file: ";TRF$
1260 LPRINT "     Old balance sheet: ";OLDBALF$
1270 LPRINT "     Old income/expense: ";OLDPAL$
1280 LPRINT "     New balance sheet: ";NEBALF$
1290 LPRINT "     New income/expense: ";NEPAL$
1300 REM SEND A FORMS FEED TO PRINTER (EPSON ASSUMED)
1310 LPRINT CHR$(12)
```

```
1320 OPEN "O",1,NEBALF$:PRINT #1,NOBAL;
1330 OPEN "O",2,NEPAL$:PRINT #2,NOPAL;
1340 FOR I=1 TO NOBAL
1350 PRINT #1,A%(I);CHR$(34);A$(I);CHR$(34);A#(I,1);A#(I,2);
1360 NEXT I:PRINT #1,0;CHR$(34);"END";CHR$(34);0;0:CLOSE 1
1370 FOR I=NOBAL+1 TO NOPAL+NOBAL
1380 PRINT #2,A%(I);CHR$(34);A$(I);CHR$(34);A#(I,1);A#(I,2);:NEXT I
1390 PRINT #2,0;CHR$(34);"END";CHR$(34);0;0
1400 CLOSE 2
1410 HOME:VTAB 3:HTAB 20:
     PRINT"*** REPORT GENERATION COMPLETED ***":PRINT:PRINT:
1420 HTAB 32:PRINT"Options Menu":PRINT:PRINT:
1430 PRINT CHR$(7):PRINT CHR$(7):
     PRINT"Select one of the following:":PRINT:PRINT:
1440 HTAB 10:PRINT" (R)e-run this program again":PRINT:
1450 HTAB 10:PRINT" (M)enu...return to the main system menu":PRINT:
1460 HTAB 10:PRINT" (T)erminate this run":PRINT:PRINT:
1470 HTAB 20:INPUT "Enter your choice (R/M/T) ";ZEB$
1480 IF LEFT$(ZEB$,1)="R" THEN GOTO 20 ELSE 1490
1490 IF LEFT$(ZEB$,1)="M" THEN RUN "MENU.BAS" ELSE 1500
1500 IF LEFT$(ZEB$,1)="T" THEN 1520 ELSE GOTO 1510
1510 PRINT CHR$(7):PRINT"****INPUT ERROR-please re-enter":GOTO 1430
1520 PRINT:HTAB 30:PRINT"***RUN TERMINATED***":END
1530 REM  INTRO SCREEN FOR REPORT
1540 VTAB 3: HTAB 23:PRINT"**** REPORT GENERATION PROGRAM ****"
1550 VTAB 8:HTAB 39:PRINT"by":VTAB 12:HTAB 27:
1560 PRINT"Steven M. Zimmerman, Ph.D.":PRINT:
1570 HTAB 33:PRINT"Leo M. Conrad":PRINT:
1580 HTAB 27:PRINT"Donald R. Smith, Ph.D.,P.E.":
1590 VTAB 20:HTAB 25:INPUT "*** (Press < CR > to Begin) ***";DUM$:HOME
1600 RETURN
1610 REM INSTRUCTION SCREEN
1620 VTAB 2: HTAB 20:
     PRINT"*** INSTRUCTIONS FOR REPORT GENERATION *** ":PRINT:PRINT:
1630 PRINT"This program generates an INCOME and BALANCE SHEET
     report using the":
1640 PRINT"current TRANS file plus the following files:":PRINT:
1650 PRINT"      * old income/expense file":
1660 PRINT"      * old balance sheet file":
1670 PRINT"      * account number file":
1680 PRINT"      * company name/address file":
1690 PRINT:PRINT"Data entry for this program follows: ":PRINT:
1700 RETURN
1710 HOME:VTAB 5:HTAB 20:
     PRINT" **** FILE MANIPULATIONS COMPLETED ****":
1720 VTAB 10:HTAB 20:PRINT"Ready to PRINT to the line printer":
1730 VTAB 15:HTAB 15:
     PRINT"  ALIGN PAPER IN PRINTER AND TURN PRINTER ON"
1740 VTAB 20:HTAB 23:INPUT "< press RETURN when ready > ";DUM$:
1750 RETURN


^@
```

# Chapter 7—INVOICE

```
10 REM: INVOICE PROGRAM
20 HOME:GOSUB 3000
30 REM 3. COMPANY.DAT RANDOM    SAVES CUSTOMER's names etc. 4.
   trans.dat  sequential Saves transactions you name it
40 ZIP$="$$###,###,###.##":ZX$="#,###,###":
   REM if you are working with non-integer quantities change ZX$=.
   .. to reflect this fact
50 ZAP$="$$##,###,###,###,###.##"
60 V=30:NF=7:NNUM=1:
70 LC$(0)="Customer's name ":LC$(1)="Company's name  ":
   LC$(2)="Number and street"
80 LC$(3)="City             ":LC$(4)="State           ":
   LC$(5)="Zip code       ":LC$(6)="First name     ":
   LC$(7)="Title           "
90 LC$(8)="Current balance "
99 WIDTH 80
120 DIM N$(V),AMOUNT(V),MONEY#(V),DATX$(V)
122 GOSUB 3100
125 PRINT:PRINT:LINE INPUT "1. BILLING DATE (XX/XX/XX) IS: ";
    DATE$:PRINT:
130 INPUT "2.  Disk Drive for Data (A,B,C,D) ";DISK$
140 PRINT:INPUT "3. Transaction file (Y/N)";NN$:
    IF LEFT$(NN$,1)="n" OR LEFT$(NN$,1)="N" THEN 200
150 PRINT:INPUT "4. Post to (I)ndividual number or (A)ccount
    rec.no";ACCNUM$
160 PRINT:HTAB 5:INPUT "Name of file";TRANSNAME$:
200 PRINT:PRINT:PRINT:
    PRINT"***** M A I N    M E N U - I N V O I C E
    P R O G R A M   *****":PRINT:PRINT:
    PRINT" Select output type:":PRINT"     (F)ILE of customers ":
    PRINT"     (M)X-80 Printer"
210 PRINT"     (P)re-printed forms":PRINT"     (O)THER printers ":
    PRINT"     (E)ND ":PRINT:HTAB 3:
    INPUT "Enter your choice (F / M / P / O / E / ) ";TYPE$
220 IF LEFT$(TYPE$,1)="E" OR LEFT$(TYPE$,1)="e" THEN
    IF LEFT$(NN$,1)="Y" OR LEFT$(NN$,1)="y" THEN ENN$="END":
    PRINT#4,0;CHR$(34);ENN$;CHR$(34);;CHR$(34);ENN$;CC$(34);0;0;0;0;
    CHR$(34):CLOSE:END ELSE END
230 IF LEFT$(TYPE$,1)="F" OR LEFT$(TYPE$,1)="f" THEN 820
240 PRINT:PRINT:HTAB 5:INPUT"New starting number (Y/N)";SNUM$
250 IF LEFT$(SNUM$,1)="n" OR LEFT$(SNUM$,1)="N" THEN
    X$=DISK$+":INVNUM.DATA":OPEN"I",#2,X$:
    INPUT #2,SN,DEBIT,SALETAX%:CLOSE 2:GOTO 270
251  REM This file contains invoice no,acct rec no and sales tax no
259 PRINT:HTAB 15:PRINT"**** Input the Following Values ***":PRINT:
260 HTAB 10:PRINT"STARTING NO.     ACCTS. REC. NO.     SALES TAX NO.":
    HTAB 10:PRINT"------------";TAB(27);"--------------";
    TAB(47);"-------------":PRINT:
```

```
261  PRINT TAB(14);:INPUT ;SN:PRINT TAB(29);:INPUT ;DEBIT:
     PRINT TAB(49);:INPUT SALETAX%
270 X$=DISK$+":NAMEX.DAT":OPEN"I",#2,X$:FOR I=0 TO 5:
    INPUT #2,COMPANY$(I):NEXT:CLOSE #2
280 PRINT:PRINT:PRINT "You must use disk file input to post to
    individual accounts in the transaction    file":
281 PRINT:INPUT "Select method-- (K)eys or (D)isk ";IM$
290 IF LEFT$(ACCNUM$,1)="I" OR LEFT$(ACCNUM$,1)="i" THEN
    IF LEFT$(IM$,1)="k" OR LEFT$(IM$,1)="K" THEN 280
300 IF LEFT$(IM$,1)="k" OR LEFT$(IM$,1)="K" THEN GOSUB 320:
    IF FXX=98 THEN 940 ELSE 390
310 IF LEFT$(IM$,1)="d" OR LEFT$(IM$,1)="D" THEN 630 ELSE 280
320 PRINT:PRINT:FOR I=0 TO NF:PRINT "Input ";LC$(I);:
    LINE INPUT LEO$(I):NEXT I:FOR I=NF+1 TO NF+NNUM:
    PRINT "Input ";LC$(I);:
321 INPUT LEO#(I):NEXT I
330 PRINT:PRINT:PRINT"Number","Title",,"Input"
340 FOR I=0 TO NF:PRINT I,LC$(I),LEO$(I):NEXT I:
    FOR I=NF+1 TO NF+NNUM:PRINT I,LC$(I),LEO#(I):NEXT I
350 PRINT:PRINT"*****  EDIT, SEARCH or CONTINUE options *****":
    INPUT "(99) to CONTINUE; (98) to  SEARCH or LINE NUMBER to edit ";
    FXX
360 IF FXX>97 THEN RETURN
370 PRINT "New information ":IF FXX>NF THEN INPUT LEO#(FXX)
    ELSE INPUT LEO$(FXX)
380 PRINT CHR$(26):GOTO 330
390 PRINT:PRINT:PRINT"*** Input INVOICE  data Section:
    To Stop, type 'END' ": I=0
400 PRINT TAB(5);:INPUT"Sales tax rate as percent (0 if none)";TAX#:
    TAX#=.01#*TAX#
410 PRINT TAB(5);:INPUT "Purchase Order No = ";PO$
415 PRINT:PRINT TAB(10);"ORDER DATE";TAB(30);"ORDER DESCRIPTION
       To end, type END <CR> END <CR>":
    PRINT TAB(10);"----------"; TAB(30);"----------------":
420 PRINT:PRINT TAB(8);:INPUT ;DATX$(I): PRINT TAB(31);:
    LINE INPUT N$(I)
430 IF N$(I)="end" OR N$(I)="END" THEN 470
440 PRINT:PRINT TAB(10);"PRICE/UNIT";TAB(30);"QUANTITY ORDERED":
    PRINT TAB(10);"----------";TAB(30);"---------------":PRINT:
445 PRINT TAB(9);:INPUT ;MONEY#(I):PRINT TAB(30);:INPUT AMOUNT(I)
450 IF N$(I)="end" OR N$(I)="END" THEN 470
460 I=I+1:GOTO 415
470 GOSUB 4000
480 SUM#=0:FOR J=0 TO I-1:SUM#=SUM#+AMOUNT(J)*MONEY#(J):
481 PRINT USING "##";J;
482 PRINT TAB(4);: PRINT USING "\        \";DATX$(J);
483 PRINT TAB(14);: PRINT USING "\                        \";N$(J);
484 PRINT TAB(38);: PRINT USING ZIP$; MONEY#(J);
485 PRINT TAB(54);: PRINT USING ZX$; AMOUNT(J);
486 PRINT TAB(64);: PRINT USING ZIP$; AMOUNT(J)*MONEY#(J)
500 NEXT
510 TAXX#=TAX#*SUM#:PRINT "          ";"     Sales tax"," ",,:
    PRINT USING ZAP$;TAXX#:SUM#=SUM#+TAXX#
```

```
520 PRINT:PRINT,,,,"        ----------------":
    PRINT,"          ",,,USING ZAP$;SUM#:
    PRINT,,,,"        ================"
530 INPUT "Input 99 if complete or number for change";CNUM
540 IF CNUM=99 THEN 560
550 PRINT:GOSUB 4000:PRINT CNUM;:
551 PRINT TAB(5);:LINE INPUT; DATX$(CNUM):PRINT TAB(15);:
    LINE INPUT; N$(CNUM):PRINT TAB(39);:INPUT; MONEY#(CNUM):
    PRINT TAB(55);: INPUT AMOUNT(CNUM)
552 GOTO 470
560 PRINT:PRINT:INPUT "Default terms (Y/N)";TERMS$
570 IF TERMS$="Y" OR TERMS$="y" THEN 600
580 PRINT:INPUT"Terms of sale";SALE$
590 GOTO 610
600 SALE$="One  percent  Ten  days,  net  thirty,  1.5%  per
    month late charge on remaining balance at end of month"
610 INPUT"<CR> when printer ready";DU$:GOTO 1300
620 REM This is section II ****************************************
630 GOSUB 790:REM  ouput from file
640 PRINT CHR$(26):PRINT:PRINT:
    PRINT" F i l e   I n p u t   M e n u ":PRINT :
    PRINT "    -1     File number"
650 PRINT "     0     Customer's name":
    PRINT "     1     Company's name":
    PRINT"     2     Number and street":PRINT "     3     CITY"
660 PRINT "     4     State":PRINT "     5     Zip Code":
    PRINT "     6     First name":INPUT "     7     Title";NOFILE:
661 IF NOFILE <-1 OR NOFILE>7 THEN 640
670 IF NOFILE=-1 THEN INPUT "File - Customer Number";K:GET 2,K:
    GOTO 740
680 INPUT "Name to be searched for";SEARCH$
690 K=0
700 K=K+1:KK=LEN(SEARCH$):IF NOFILE=-1 THEN GET 2,K:GOTO 740
710 GET 2,K:K$=CCUST$(NOFILE):IF LEFT$(K$,KK)=SEARCH$ THEN 740
720 IF LEFT$(K$,3)="END" THEN
    INPUT "Cannot find file Input to try again";DU$:CLS:
    CLOSE 2:GOTO 200
730 GOTO 700
740 FOR J=0 TO NF:LEO$(J)=CCUST$(J):NEXT J:FOR J=NF+1 TO NF+NNUM:
    LEO#(J)=CVD(CCUST$(J)):NEXT J
750 GOSUB 330:IF FXX=98 THEN 700
760 CLOSE 2:GOTO 390
770 REM This is section III ***************************************
780 REM  open customer.data file
790 X$=DISK$+":COMPANY.DAT"
800 OPEN "R",2,"B:COMPANY.DAT",111:FIELD #2,12 AS CCUST$(0),
    25 AS CCUST$(1),25 AS CCUST$(2),15 AS CCUST$(3),3 AS CCUST$(4),
    5 AS CCUST$(5),8 AS CCUST$(6),10 AS CCUST$(7),8 AS CCUST$(8):RETURN
810 REM customer's name (12), company's name (25), number and street
    (25), city (15), state (3), zip (5), first name (8)
811 REM title(10), remaining bal (8)
812 REM number fields in file NF=7 (eight---zero position is used
    and NNUM=1 a single number
820 GOSUB 790:REM lines 700 through 950 are for file mainentance
```

```
830 PRINT CHR$'26):K$="":PRINT:PRINT:
    PRINT" F i l e   M a i n t e n a n c e   M e n u "
840 PRINT:PRINT"     (A)DD":PRINT"     (B)ROWSE the Files":
    PRINT"     (D)elete file":PRINT"     (E)ND  ":
    PRINT "     (S)TART   NEW FILE"
850 PRINT"     (U)PDATE  existing file":PRINT:
    INPUT"Enter your choice ";FILE$
860 IF LEFT$(FILE$,1)="E" OR LEFT$(FILE$,1)="e" THEN CLOSE 2:
    PRINT CHR$(26):GOTO 200
870 IF LEFT$(FILE$,1)="A" OR LEFT$(FILE$,1)="a" THEN GOSUB 320:
    GOTO 920
880 IF LEFT$(FILE$,1)="D" OR LEFT$(FILE$,1)="d" THEN 970
890 IF LEFT$(FILE$,1)="S" OR LEFT$(FILE$,1)="s" THEN 1140
900 IF LEFT$(FILE$,1)="U" OR LEFT$(FILE$,1)="u" THEN 970
905 IF LEFT$(FILE$,1)="B" OR LEFT$(FILE$,1)="b" THEN GOSUB 2500
910 GOTO 830
920 K=0
930 K=K+1
940 GET 2,K:K$=CCUST$(0):IF LEFT$(K$,3)="END" OR LEFT$(K$,3)="OUT"
    THEN 950 ELSE 930
950 FOR J=0 TO NF:LSET CCUST$(J)=LEO$(J):NEXT J:
    FOR J=NF+1 TO NF+NNUM:LSET CCUST$(J)=MKD$(LEO#(J)):NEXT J:
    PUT 2,K:
951 IF LEFT$(K$,3)="END" THEN FOR J=0 TO NF:LSET CCUST$(J)="END":
    NEXT J:FOR J=NF+1 TO NF+NNUM:LSET CCUST$(J)=MKD$(9999999999#)
    :NEXT J:PUT 2,K+1
960 GOTO 830
970 PRINT :PRINT :
    PRINT "I N V O I C E   P R O G R A M   S e a r c h   M e n u":
    PRINT:PRINT:
980 HTAB 4:PRINT"-1     By File Number":HTAB 5:
    PRINT"0     Customer' Name":HTAB 5:
    PRINT"1     Company's Name":HTAB 5:PRINT"2     Number and Street"
990 HTAB 5:PRINT"3     City":HTAB 5:PRINT"4     State":HTAB 5:
    PRINT"5     Zip Code":HTAB 5:PRINT"6     First Name":
    HTAB 5:PRINT"7     Title":HTAB 5:PRINT"8     Current Balance":
994 PRINT:INPUT "Enter your CHOICE "; NOFILE:
    IF NOFILE <-1 OR NOFILE >8 THEN 970
1000 IF NOFILE=-1 THEN INPUT "File - Customer Number";K:
     GET 2,K:GOTO 1080
1010 IF NOFILE=8 THEN 1170
1020 INPUT "Name to be searched for";SEARCH$
1030 K=0
1040 K=K+1:KK=LEN(SEARCH$):IF NOFILE=-1 THEN GET 2,K:GOTO 1080
1050 GET 2,K:K$=CCUST$(NOFILE):PRINT K$:IF LEFT$(K$,KK)=SEARCH$
     THEN 1080
1060 IF LEFT$(K$,3)="END" THEN
     INPUT "Cannot find file Input to try again";DU$:GOTO 830
1070 GOTO 1040
1080 FOR J=0 TO NF:LEO$(J)=CCUST$(J):NEXT J:FOR J=NF+1 TO NF+NNUM:
     LEO#(J)=CVD(CCUST$(J)):NEXT J
1090 GOSUB 330
1100 IF FXX=98 THEN 1040
1110 IF LEFT$(FILE$,1)="d" OR LEFT$(FILE$,1)="D" THEN 1130
1120 GOTO 950
```

```
1130 FOR J=0 TO NF:LSET CCUST$(J)="OUT":NEXT J:
     FOR J=NF+1 TO NF+NNUM:LSET CCUST$(J)=MKD$(999999999#):NEXT J:
     PUT 2,K:GOTO 830
1140 PRINT :PRINT :PRINT:
     INPUT "Are you sure you wish to continue type  completely ...
     YES or NO";YY$:IF YY$<>"YES" THEN CLOSE 2:GOTO 200
1150 FOR I=0 TO NF:LSET CCUST$(I)="END":NEXT I:FOR J=NF+1 TO NF+NNUM:
     LSET CCUST$(J)=MKD$(999999999#):NEXT J:PUT 2,1:PUT 2,2
1160 GOTO 830
1170 PRINT CHR$(26):PRINT "Value search alternatives":
     PRINT "     EQUAL":PRINT "     GREATER":
     PRINT "     LESS";:INPUT VALUE$
1180 INPUT "Value ";VALUE#:IF VALUE# <0 THEN 1170
1190 K=0
1200 K=K+1:GET 2,K:DD#=CVD(CCUST$(NOFILE)):
     IF LEFT$(VALUE$,1)="E" OR LEFT$(VALUE$,1)="e" THEN
     IF DD#=VALUE# THEN 1250
1210 IF LEFT$(VALUE$,1)="G" OR LEFT$(VALUE$,1)="g" IF
     DD#>VALUE# THEN 1250
1220 IF LEFT$(VALUE$,1)="L" OR LEFT$(VALUE$,1)="l" IF
     DD#<VALUE# THEN 1250
1230 IF DD#<99999999# THEN 1200
1240 INPUT "Cannot find file Input to try again";DU:GOTO 830
1250 FOR J=0 TO NF:LEO$(J)=CCUST$(J):NEXT J:
     FOR J=NF+1 TO NF+NNUM:LEO#(J)=CVD(CCUST$(J)):NEXT J
1260 GOSUB 330
1270 IF FXX=98 THEN 1200
1280 GOTO 830
1290 REM This is section IV ****************************************
1300 IF LEFT$(TYPE$,1)="p" OR LEFT$(TYPE$,1)="P" THEN 1930:
     REM lines 1320 through 1600 are for MX80 printer output
1310 IF LEFT$(TYPE$,1)="o" OR LEFT$(TYPE$,1)="O" THEN 1640
1320 LPRINT:LPRINT TAB(5)CHR$(14)COMPANY$(0)
1330 LPRINT TAB(5)CHR$(27)CHR$(71)"Accounts Receivable"
1340 LPRINT CHR$(15)
1350 LPRINT TAB(9)CHR$(27)CHR$(72)COMPANY$(1)
1360 LPRINT TAB(9)COMPANY$(2);",  ";COMPANY$(3);" ";COMPANY$(4):LPRINT
1370 LPRINT TAB(9);"Phone: ";COMPANY$(5)
1380 LPRINT
1390 LPRINT CHR$(18)
1400 LPRINT "   ";CHR$(42);:FOR J=1 TO 39:LPRINT CHR$(42);:
     NEXT J:LPRINT CHR$(42);
1410 LPRINT TAB(47) "Invoice: ";SN;"  Date: ";DATE$
1420 LPRINT "   ";CHR$(42);:
     LPRINT USING "\                                          \";
     LEO$(0);:LPRINT TAB(44) CHR$(42)
1430 LPRINT "   ";CHR$(42);:
     LPRINT USING "\                                            \";
     LEO$(1);:LPRINT TAB(44)CHR$(42);:
     LPRINT TAB(47) "Purchase Order #: ";PO$
1440 LPRINT "   ";CHR$(42);:
     LPRINT USING "\                                          \";LEO$(2);:
     LPRINT TAB(44)CHR$(42)

1450 LPRINT "   ";CHR$(42);:
```

```
     LPRINT USING "\             \, ";LEO$(3);:
     LPRINT USING "\\  ";LEO$(4);:LPRINT  USING "\   \";LEO$(5);:
1451 LPRINT TAB(44) CHR$(42)
1460 LPRINT "   ";CHR$(42);:FOR J=1 TO 39:LPRINT CHR$(42);:NEXT J:
     LPRINT CHR$(42)
1470 LPRINT:LPRINT:LPRINT:LPRINT
1480 LPRINT CHR$(42);:FOR J=1 TO 73:LPRINT CHR$(42);:NEXT J:
     LPRINT CHR$(42)
1490 LPRINT CHR$(42);:LPRINT "Date";TAB(12);"DESCRIPTION";TAB(45);
     "Price  Quantity  SUB-TOTAL";TAB(75);CHR$(42)
1500 LPRINT CHR$(42);:FOR J=1 TO 73:LPRINT CHR$(42);:NEXT J:
     LPRINT CHR$(42)
1510 FOR J=0 TO I-1:LPRINT CHR$(42);USING "\       \";DATX$(J);
1516 LPRINT USING "\                    \";N$(J);
1517 LPRINT USING ZIP$;MONEY#(J);
1518 LPRINT USING ZX$;AMOUNT(J);
1519 LPRINT USING ZIP$;MONEY#(J)*AMOUNT(J);:LPRINT TAB(75) CHR$(42)
1520 GOSUB 2330
1530 NEXT J
1540 LPRINT CHR$(42);"          ";
     "Sales tax          "," "," ";:LPRINT USING ZIP$;TAXX#;:
     LPRINT TAB(75)CHR$(42)
1550 LPRINT CHR$(42);:FOR J=1 TO 73:LPRINT CHR$(42);:NEXT J:
     LPRINT CHR$(42)
1560 LPRINT:LPRINT TAB(47)"Total      ";USING ZIP$;SUM#:
     LPRINT TAB(60);:FOR J=1 TO 16:LPRINT CHR$(61);:NEXT J
1570 LPRINT:LPRINT
1580 LPRINT CHR$(15):LPRINT "**** TERMS ****":LPRINT:
1590 LPRINT SALE$:LPRINT CHR$(27)CHR$(72)
1600 LPRINT CHR$(18):LPRINT CHR$(12)
1605 REM END OF THE EPSON PRINT SECTION  **************************
1610 GOSUB 2160
1620 GOTO 200
1630 REM This is section V ****************************************
1640 REM lines are to output on regular printer using plane paper
1650 LPRINT TAB(6)COMPANY$(0)
1660 LPRINT TAB(6)"Accounts Receivable"
1670 LPRINT TAB(6)COMPANY$(1)
1680 LPRINT TAB(6)COMPANY$(2);",";"  ";COMPANY$(3); ";COMPANY$(4):
     LPRINT
1690 LPRINT TAB(6)COMPANY$(5)
1700 LPRINT:LPRINT
1710 LPRINT "    ";:FOR J=1 TO 41:LPRINT "*";:NEXT
1720 LPRINT TAB(48) "Invoice:  ";SN;"   Date:  ";DATE$
1730 LPRINT"    ";:LPRINT "*";:LPRINT USING "\
1731 LPRINT "   Purchase Order #: ";PO$
1740 LPRINT"    ";:LPRINT "*";:LPRINT USING "\
1750 LPRINT "    *";:
     LPRINT USING "\                                    \";LEO$(2);:
     LPRINT TAB(45)"*"
1760 LPRINT "    *";:LPRINT USING "\             \, ";LEO$(3);:
     LPRINT USING "\\  ";LEO$(4);:LPRINT USING "\   \";LEO$(5);:
1761 LPRINT TAB(45)"*"
1770 LPRINT "    *";:FOR J=1 TO 39:LPRINT "*";:NEXT J:LPRINT"*"
1780 LPRINT:LPRINT:LPRINT:LPRINT
```

```
1790 FOR J=1 TO 75:LPRINT "*";:NEXT J:LPRINT
1800 LPRINT "*";:LPRINT"Date";TAB(12)"Description";
     TAB(45)"Price   Quantity   Sub-total";TAB(75)"*"
1810 FOR J=1 TO 75:LPRINT "*";:NEXT J:LPRINT
1820 FOR J=0 TO I-1:LPRINT "*";USING "\       \";DATX$(J);:
     LPRINT USING "\                    \";N$(J);:
1821 LPRINT USING ZIP$;MONEY#(J);:LPRINT USING ZX$;AMOUNT(J);:
     LPRINT USING ZIP$;MONEY#(J)*AMOUNT(J);:
1822 LPRINT TAB(75) "*"
1830 GOSUB 2330
1840 NEXT J
1850 LPRINT "*";"          ";"Sales tax          "," "," ";:
     LPRINT USING ZIP$;TAXX#;:LPRINT TAB(75)"*"
1860 FOR J=1 TO 74:LPRINT "*";:NEXT J:LPRINT "*"
1870 LPRINT TAB(47) "Total      ";USING ZIP$;SUM#:LPRINT TAB(60);:
     FOR J=1 TO 16:LPRINT "*";:NEXT J
1880 LPRINT:LPRINT
1890 LPRINT SALE$:LPRINT
1900 GOSUB 2160
1910 GOTO 200
1920 REM This is section VI ************************************
1930 LEFT=10
1940 LPRINT TAB(LEFT+47)DATE$;" ";SN
1950 REM replace the next 6 statements with 8 LPRINTs if
     stationary has printed name
1960 LPRINT TAB(LEFT+3)COMPANY$(0)
1970 LPRINT TAB(LEFT+3)"Accounts Receivable"
1980 LPRINT TAB(LEFT+3)COMPANY$(1)
1990 LPRINT TAB(LEFT+3)COMPANY$(2);",";COMPANY$(3);Y$(4):LPRINT
2000 LPRINT TAB(LEFT+3)COMPANY$(5)
2010 LPRINT:LPRINT:LPRINT
2020 FOR J=0 TO 3:LPRINT TAB(LEFT+3) LEO$(J);:NEXT
2030 LPRINT "    ";LEO$(4);"   ";LEO$(5):LPRINT
2040 LPRINT
2050 FOR J=0 TO I-1:LPRINT TAB(LEFT-4);USING "\       \";DATX$(J);:
     LPRINT USING "\                    \";N$(J);:
2051 LPRINT USING ZIP$;MONEY#(J);:LPRINT USING ZX$;AMOUNT(J);:
     LPRINT USING ZIP$;MONEY#(J)*AMOUNT(J);
2060 GOSUB 2330
2070 NEXT J
2080 LPRINT TAB(LEFT-4);"          ";"Sales tax          "," "," ";:
     LPRINT "   "; USING ZIP$;TAXX#
2090 LPRINT TAB(53+LEFT)"-------------"
2100 LPRINT TAB(38+LEFT)"Total      ";USING ZIP$;SUM#:
     LPRINT TAB(51+LEFT);:FOR J=1 TO 16:LPRINT "=";:NEXT J
2110 LPRINT :LPRINT
2120 LPRINT  SALE$:LPRINT
2130 GOSUB 2160
2140 GOTO 200
2150 REM this subroutine posts the information to the transaction file
     as defined and to the customer's file
2160 SN=SN+1:X$=DISK$+":INVNUM.DAT":OPEN "O",#2,X$:
     PRINT #2,SN;DEBIT;SALETAX%:CLOSE 2

2170 SUM#=SUM#-TAXX#:DOLLAR#=SUM#:GOSUB 2310:SUM#=DOLLAR#:
```

```
     DOLLAR#=TAXX#:GOSUB 2310:TAXX#=DOLLAR#
2180 IF LEFT$(NN$,1)="N" OR LEFT$(NN$,1)="n" THEN RETURN
2190 SN%=SN
2200 K%=K+DEBIT:IF LEFT$(ACCNUM$,1)="A" OR LEFT$(ACNUM$,1)="a"
     THEN K%=DEBIT
2210 INPUT "INPUT sales income account number";SALEACCT%
2220 REM  We must debit accounts receivable and credit the
     source of sales
2230 X$="Invoice":Y$="ISSUED":PRINT #4,SN%;CHR$(34),DATE$;CHR$(34);
     CHR$(34);X$;CHH$(34);K%;SALEACCT%;SUM#;SUM#;CHR$(344Y$;CHR$(34)
2239 X$="Sales Tax"
2240 IF TAXX#>0# THEN PRINT #4,SN%;CHR$(34);DATE$;CHR$(34);CHR$(34);
     X$;CHR$(34);KKSALETAX%;TAXX#;TAXX#;CHR$(344Y$;CHR$(34)
2250 IF LEFT$(ACCNUM$,1)="A" OR LEFT$(ACCNUM$,1)="a" THEN RETURN
2260 SUM#=SUM#+TAXX#+CVD(CCUST$(8)):GOSUB 790
2270 FOR J=0 TO NF-1:LSET CCUST$(J)=LEO$(J)
2280 NEXT J:LSET CCUST$(8)=MKD$(SUM#):PUT 2,K:CLOSE 2
2290 RETURN
2300 REM the transaction file includes transaction # DATE#
     Description$ acct debit/credit amt deb/cr and Clear$
2310 XXXX#=INT(DOLLAR#*100#)/100#:DIFF#=DOLLAR#-XXXX#:
     IF DIFF#>5.000000000000003D-03 THEN XXXX#=XXXX#+.01#
2320 DOLLAR#=XXXX#:RETURN:REM this subroutine corrects for round-off
     errors before storing informaton in data file
2330 KLIP=INT(J/42)*42:IF KLIP=J AND J>0 THEN LPRINT CHR$(12)
2340 RETURN
2500 REM BROWSE CUSTOMER FILE SUBROUTINE
2510 HOME:VTAB 3:HTAB 10:
     PRINT"B R O W S E   C U S T O M E R   F I L E   S E C T I O N"
2520 PRINT:PRINT:HTAB 15:PRINT"*** Browse File options ***"
2530 HTAB 10: PRINT"(1) - SCREEN DISPLAY ONLY":HTAB 10:
     PRINT"(2) - LINE PRINTER SUMMARY ONLY":HTAB 10:
     PRINT"(3) - RETURN TO MAIN MENU":PRINT:PRINT:
2540 HTAB 10: INPUT "Enter your choice (1/2/3) ";XX$
2550 IF XX$="1" THEN GOTO 2580 ELSE 2560
2560 IF XX$="2" THEN GOTO 2720 ELSE 2570
2570 IF XX$="3" THEN HOME: VTAB 8: RETURN ELSE GOTO 2520
2580 REM SCREEN ONLY BROWSE OPTION
2590 HOME:VTAB 2: PRINT"****  Beginning File Listing Beginning
     with Record No. 1 ***":PRINT:PRINT:GOTO 2630
2600 FOR M%=1 TO 80: PRINT CHR$(45);:NEXT M%:
2610 PRINT"File No";TAB(10);"Cust. L. Name";TAB(30);
     "Company's Name";TAB(56);"City";TAB(75);"State"
2620 FOR M%=1 TO 80:PRINT CHR$(45);:NEXT M%:RETURN
2630 GOSUB 2600:K=0:KC%=0
2640 K=K+1
2650 GET #2,K:K$=CCUST$(0)
2660 FOR J=0 TO NF: LEO$(J)=CCUST$(J):NEXT J:
     FOR J = NF+1 TO NF+NNUM:LEO#(J)=CVD(CCUST$(J)):NEXT J
2670 PRINT TAB(3);K;TAB(10);LEO$(0);TAB(24);LEO$(1);TAB(50);
     LEO$(3);TAB(76);LEO$((
2680 IF LEFT$(LEO$(0),3)="END" THEN GOTO 2690 ELSE 2700
2690 PRINT:PRINT:HTAB 25:INPUT "Press <CR> to Continue ";DUM$:
     GOTO 2510
2700 KC%=KC%+1: IF KC%=15 THEN KC%=0:PRINT:PRINT:HTAB 25:
```

```
    INPUT "Press <CR> to Resume Viewing ";DUM$:
2710 GOTO 2640
2720 REM LINE PRINTER ROUTINE FOR BROWSE FILE OPTION
2730 HOME:VTAB 3:
    PRINT"* * * * L I N E   P R I N T E R   O P T I O N
       for  B R O W S E":PRINT:PRINT:
2740 PRINT"  ALIGN paper in printer to top of form and TURN
    printer ON  "
2750 PRINT:PRINT: HTAB 20:INPUT " Press <CR> when ready to Begin ";DUM$
2760 LPRINT:LPRINT: LPRINT "***** HARDCOPY LISTING OF THE CUSTOMER FILE
    FROM THE INVOICE PROGRAM *****":LPRINT:LPRINT:
    LPRINT "Date of Run: ";DATE$:LPRINT:LPRINT:
2770 K=0:KC%=0
2780 K=K+1
2790 LPRINT TAB(30);"FILE NO. ";K:LPRINT:
2800 GET #2,K:K$=CCUST$(0):  FOR J=0 TO NF: LEO$(J)=CCUST$(J):
    NEXT J:FOR J=NF+1 TO NF+NNUM:LEO#(J)=CVD(CCUST$(J)):NEXT J:
2810 FOR I=0 TO NF:LPRINT TAB(10);LEO$(I):NEXT I:
2812 FOR I=NF+1 TO NF+NNUM:LPRINT TAB(10);LEO#(I):NEXT I
2820 LPRINT: LPRINT:LPRINT:
2830 IF LEFT$(LEO$(0),3)="END" THEN GOTO 2840 ELSE 2832
2832 KC%=KC%+1:IF KC%=4 THEN LPRINT CHR$(12):GOTO 2780
2834 GOTO 2780
2840 LPRINT CHR$(12):GOTO 2510
3000 REM PROMO SCREEN FOR INVOICE
3010 VTAB 3:HTAB 23:PRINT"**** INVOICE GENERATION PROGRAM ****"
3020 VTAB 8: HTAB 39:PRINT"by":VTAB 12:HTAB 27:
3030 PRINT"Steven M. Zimmerman Ph.D.":PRINT:
3040 HTAB 33: PRINT"Leo M. Conrad":PRINT:
3050 HTAB 27:PRINT"Donald R. Smith Ph.D., P.E.":
3060 VTAB 20:HTAB 28:INPUT "( Press < CR > to Begin )";DUM$:HOME:
3070 RETURN
3100 REM SCREEN DIRECTIONS FOR USER
3110 VTAB 2:PRINT" *** USER DIRECTIONS: ":PRINT:PRINT:
3120 PRINT"THIS PROGRAM IS DESIGNED TO MAINTAIN A FILE OF CUSTOMERS
    FROM WHICH"
3130 PRINT"INVOICES MAY BE PRODUCED, OR TO PRODUCE INVOICES FROM
    THE KEYBOARD"
3140 PRINT"THE PROGRAM OPERATES FROM A SERIES OF MENUS.
    THE FILING SYSTEM USED"
3150 PRINT"BY THE PROGRAM IS A RANDOM ACCESS SYSTEM.
    IT IS SUGGESTED THAT YOUR"
3160 PRINT"CUSTOMER FILES BE MAINTAINED ON A SEPARATE DISK FROM
    THE ACCOUNTING"
3170 PRINT"PACKAGE.  HOWEVER, PLACE A COPY OF 'NAMEX.DAT' FROM
    THE SETUP"
3172 PRINT"PROGRAM ON THIS DISK.  THE PIP UTILITY CAN BE USED TO
    COPY THIS FILE"
3174 PRINT"FROM THE ACCT. DATA DISK TO THE INVOICE DATA DISK.
    IF YOU HAVE A"
3176 PRINT"2-DRIVE SYSTEM, USE DRIVE B FOR INVOICE DATA."
3180 RETURN


4000 PRINT:PRINT:PRINT"No.";TAB(6);"Date";TAB(15);
```

```
     "Item Description";TAB(38);"Price/Unit";TAB(54);"Quantity";
     TAB(67);"Extended Total"
4010 FOR KKK=1 TO 80:PRINT CHR$(45);:NEXT KKK:PRINT:RETURN
5000 END

^@
```

# Chapter 8—MAILLIST APPLE CP/M

```
10 REM "MAILLIST"
20 WIDTH 80:HOME:GOSUB 3000
30 REM THIS PROGRAM USES ONE FILE:
40 REM    COMPANY.DAT OR SUPPLY.DAT NAMES
50 FAMOUNT=400
60 DIM ORDER(FAMOUNT),WORK$(FAMOUNT):REM UP TO 400 NAMES IN THE FILE
70 ST$="\      \"
80 KKK$="\           \ "
90 KKX$="###,###.## "
100 GOSUB 1710
110 V=30:NF=7:NNUM=1:REM NF IS THE NUMBER OF STRING$ IN THE
    CUSTOMER'S FILE
120 LC$(0)="LAST NAME       "
130 LC$(1)="COMPANY'S NAME  ":LC$(2)="NUMBER AND STREET"
140 LC$(3)="CITY            ":LC$(4)="STATE           "
150 LC$(5)="ZIP CODE        "
160 LC$(6)="FIRST NAME      ":LC$(7)="TITLE           "
170 LC$(8)="CURRENT BALANCE "
210 INPUT "1. (C)ustomer's file or (S)upplier's file ";CCUSUP$
220 PRINT:PRINT
230 INPUT "2. Disk (A,B,C,D)";DISK$:GOSUB 580
240 PRINT :PRINT "OUTPUT MENU":PRINT "     (E)ND":
    PRINT "     (F)ILES"
250 PRINT"     (L)ABELS ":INPUT "     (M)AILING ENVELOPES ";TYPE$
260 IF LEFT$(TYPE$,1)="E" OR LEFT$(TYPE$,1)="E" THEN CLOSE:RUN "MENU"
270 IF LEFT$(TYPE$,1)="L" OR LEFT$(TYPE$,1)="L" THEN GOSUB 1450
280 IF LEFT$(TYPE$,1)="M" OR LEFT$(TYPE$,1)="M" THEN GOSUB 1450
290 IF LEFT$(TYPE$,1)="F" OR LEFT$(TYPE$,1)="F" THEN 640
300 GOTO 240
310 REM SUBROUTINE FOR FILE INPUT
320 IF EIGHT=50 THEN WIDTH 80 ELSE WIDTH 80
325 PRINT:PRINT:FOR I=0 TO NF:PRINT "INPUT ";LC$(I);
330 INPUT LEO$(I):NEXT I
340 FOR I=NF+1 TO NF+NNUM:PRINT "INPUT ";LC$(I);:INPUT LEO#(I):NEXT I
350 PRINT:PRINT:PRINT"NUMBER","TITLE",,"INPUT"
360 FOR I=0 TO NF:PRINT I,LC$(I),LEO$(I):NEXT I
370 FOR I=NF+1 TO NF+NNUM:PRINT I,LC$(I),LEO#(I):NEXT I
380 INPUT "99 CONTINUE, 98 SEARCH, OR NUMBER TO CHANGE";FXX
390 IF FXX>97 THEN RETURN
400 PRINT "NEW INFORMATION ":IF FXX>NF THEN INPUT LEO#(FXX) ELSE
    INPUT LEO$(FXX)
410 HOME:GOTO 350
420 REM FILE OUTPUT  ***********************************************
430 HOME:PRINT:PRINT" F I L E   I N P U T   M E N U "
440 TL=7
450 GOSUB 1860
460 IF NOFILE=-1 THEN INPUT "FILE - CUSTOMER NUMBER";K:
    GET 2,K:GOTO 530
470 INPUT "NAME TO BE SEARCHED FOR";SEARCH$
480 K=0
490 K=K+1:KK=LEN(SEARCH$):IF NOFILE=-1 THEN GET 2,K:GOTO 530
500 GET 2,K:K$=CCUST$(NOFILE):IF LEFT$(K$,KK)=SEARCH$ THEN 530
```

```
510 IF LEFT$(K$,3)="END" THEN
    INPUT "CANNOT FIND FILE INPUT TO TRY AGAIN";DU$:HOME:GOTO 240
520 GOTO 490
530 FOR J=0 TO NF:LEO$(J)=CCUST$(J):NEXT J
540 FOR J=NF+1 TO NF+NNUM:LEO#(J)=CVD(CCUST$(J)):NEXT J
550 GOSUB 350:IF FXX=98 THEN 490
560 RETURN
570 REM  OPEN CUSTOMER.DATA FILE -- MAINTENANCE SECTION **************
580 X$=DISK$+":COMPANY.DAT"
590 IF LEFT$(CUSSUP$,1)="S" OR LEFT$(CUSSUP$,1)="S" THEN
    X$=DISK$+":SUPPLY.DAT"
600 OPEN "R",2,X$,111
610 FIELD #2,12 AS CCUST$(0),25 AS CCUST$(1),25 AS CCUST$(2),
    15 AS CCUST$(3),3 AS CCUST$(4),5 AS CCUST$(5),8 AS CCUST$(6),
    10 AS CCUST$(7),8 AS CCUST$(8)
620 RETURN
630 REM CUSTOMER'S NAME (12), COMPANY'S NAME (25),
    NUMBER AND STREET (25),
631 REM CITY (15), STATE (3), ZIP (5), FIRST NAME (8), TITLE (10),
632 REM REMAINING BALANCE (8):REM NUMBER FIELDS IN FILE NF=7
633 REM (EIGHT---- THE ZERO POSITION IS USED AND NNUM=1 A
    SINGLE NUMBER)
640 WIDTH 80:HOME:K$="":HTAB 14:PRINT:PRINT:
    PRINT" F I L E   M A I N T E N A N C E   M E N U "
650 PRINT:
    PRINT"      (A)DD NEW CUSTOMER":
    PRINT "      (E)ND CYCLE":PRINT "      (D)ELETE FILE"
660 PRINT"      (L)IST"
670 PRINT"      (P)RINT LIST"
680 PRINT"      (O)RDER"
690 PRINT"      (S)TART NEW FILE"
700 PRINT"      (U)PDATE AN EXISTING FILE":INPUT"  ";FILE$:
    IF EIGHT=50 THEN WIDTH 80 ELSE WIDTH 80
710 IF LEFT$(FILE$,1)="E" OR LEFT$(FILE$,1)="E" THEN HOME:GOTO 240
720 IF LEFT$(FILE$,1)="L" OR LEFT$(FILE$,1)="L" THEN 1910
730 IF LEFT$(FILE$,1)="P" OR LEFT$(FILE$,1)="P" THEN 2000
740 IF LEFT$(FILE$,1)="O" OR LEFT$(FILE$,1)="O" THEN GOSUB 1670
750 IF LEFT$(FILE$,1)="A" OR LEFT$(FILE$,1)="A" THEN
    GOSUB 320:GOTO 800
760 IF LEFT$(FILE$,1)="D" OR LEFT$(FILE$,1)="D" THEN 880
770 IF LEFT$(FILE$,1)="S" OR LEFT$(FILE$,1)="S" THEN 1080
780 IF LEFT$(FILE$,1)="U" OR LEFT$(FILE$,1)="U" THEN 880
790 GOTO 640
800 K=0
810 K=K+1
820 GET 2,K:K$=CCUST$(0)
830 IF LEFT$(K$,3)="END" OR LEFT$(K$,3)="OUT" THEN 840 ELSE GOTO 810
840 FOR J=0 TO NF:LSET CCUST$(J)=LEO$(J):NEXT
850 FOR J=NF+1 TO NF+NNUM:LSET CCUST$(J)=MKD$(LEO#(J)):NEXT J:PUT 2,K
860 IF LEFT$(K$,3)="END" THEN FOR J=0 TO NF:LSET CCUST$(J)="END":
    NEXT J:FOR J=NF+1 TO NF + NNUM:LSET CCUST$(J)=MKD$(9999999999#):
    NEXT J: PUT 2,K+1
870 GOTO 640
880 REM DELETE AND UPDATE ROUTINES
890 TL=8
```

```
900 GOSUB 1860
910 IF NOFILE=-1 THEN INPUT "FILE - CUSTOMER NUMBER";K:
    GET 2,K:GOTO 990
920 IF NOFILE=8 THEN 1150
930 INPUT "NAME TO BE SEARCHED FOR";SEARCH$
940 K=0
950 K=K+1:KK=LEN(SEARCH$):IF NOFILE=-1 THEN GET 2,K:GOTO 990
960 GET 2,K:K$=CCUST$(NOFILE):PRINT K$:IF LEFT$(K$,KK)=SEARCH$
    THEN 990
970 IF LEFT$(K$,3)="END" THEN INPUT "CANNOT FIND FILE INPUT TO
    TRY AGAIN";DU$:GOTO 640
980 GOTO 950
990 FOR J=0 TO NF:LEO$(J)=CCUST$(J):NEXT J
1000 FOR J=NF+1 TO NF+NNUM:LEO#(J)=CVD(CCUST$(J)):NEXT J
1010 GOSUB 350
1020 IF FXX=98 THEN 950
1030 IF LEFT$(FILE$,1)="D" OR LEFT$(FILE$,1)="D" THEN 1050
1040 GOTO 840
1050 FOR J=0 TO NF:LSET CCUST$(J)="OUT":NEXT J
1060 FOR J=NF+1 TO NF+NNUM:LSET CCUST$(J)=MKD$(9999999999#):NEXT J
1070 PUT 2,K:GOTO 640
1080 WIDTH 80:PRINT :PRINT :PRINT
1090 INPUT "ARE YOU SURE YOU WISH TO CONTINUE TYPE
     COMPLETELY ... YES OR NO";YY$
1100 IF YY$<>"YES" THEN WIDTH 80:CLOSE 2:GOTO 20
1110 FOR I=0 TO NF:LSET CCUST$(I)="END":NEXT I
1120 FOR J=NF+1 TO NF+NNUM:LSET CCUST$(J)=MKD$(9999999999#)
1130 NEXT J:PUT 2,1:PUT 2,2
1140 WIDTH 80:GOTO 640
1150 HOME:PRINT "VALUE SEARCH ALTERNATIVES":PRINT "     EQUAL"
1160 PRINT "     GREATER":PRINT "     LESS";:INPUT VALUE$
1170 INPUT "VALUE ";VALUE#:IF VALUE# <0 THEN 1150
1180 K=0
1190 K=K+1:GET 2,K:DD#=CVD(CCUST$(NOFILE))
1200 IF LEFT$(VALUE$,1)="E" OR LEFT$(VALUE$,1)="E" THEN
     IF DD#=VALUE# THEN 1250
1210 IF LEFT$(VALUE$,1)="G" OR LEFT$(VALUE$,1)="G" THEN
     IF DD#>VALUE# THEN 1250
1220 IF LEFT$(VALUE$,1)="L" OR LEFT$(VALUE$,1)="L" THEN
     IF DD#<VALUE# THEN 1250
1230 IF DD#<99999999# THEN 1190
1240 INPUT "CANNOT FIND FILE INPUT TO TRY AGAIN";DU$:GOTO 640
1250 FOR J=0 TO NF:LEO$(J)=CCUST$(J):NEXT J
1260 FOR J=NF+1 TO NF+NNUM:LEO#(J)=CVD(CCUST$(J)):NEXT J
1270 GOSUB 350
1280 IF FXX=98 THEN 1190
1290 GOTO 640
1300 REM THIS IS MAILING ENVELOPES **********************************
1310 LPRINT:LPRINT:LPRINT:REM YOU MUST CONTROL
1320 LEFT=40:REM YOU MUST CONTROL
1330 FOR J=0 TO 3:LPRINT TAB(LEFT+3) LEO$(J);:NEXT J
1340 LPRINT ",  ";LEO$(4);"  ";LEO$(5):LPRINT
1350 LPRINT
1360 RETURN
1370 REM THIS IS PRINTS LABELS **************************************
1380 LEFT=5:REM YOU MUST CONTROL
```

```
1390 LPRINT:LPRINT:LPRINT:REM YOU MUST CONTROL
1400 FOR J=0 TO 3:LPRINT TAB(LEFT+3) LEO$(J);:NEXT J
1410 LPRINT ",  ";LEO$(4);"  ";LEO$(5):LPRINT
1430 RETURN
1440 REM TO DEVELOP PRINTING CRITERIA ****************************
1450 PRINT :PRINT
1460 PRINT "P R I N T      M E N U"
1470 TL=7
1480 GOSUB 1860
1490 IF NOFILE=-1 THEN INPUT "STARTING- CUSTOMER NUMBER";K:GOTO 1580
1500 INPUT "NAME TO BE SEARCHED FOR";SEARCH$
1510 K=0
1520 K=K+1:KK=LEN(SEARCH$)
1530 GET 2,K:GOSUB 1650
1540 K$=CCUST$(NOFILE):PRINT K$
1550 IF LEFT$(K$,KK)=SEARCH$ THEN GOSUB 1620
1560 IF LEFT$(LEO$(0),3)="END" THEN RETURN
1570 GOTO 1520
1580 GET 2,K:GOSUB 1650:PRINT K
1590 IF LEFT$(LEO$(1),3)="END" THEN RETURN
1600 GOSUB 1620
1610 K=K+1:GOTO 1580
1620 IF LEFT$(TYPE$,1)="L" OR LEFT$(TYPE$,1)="L" THEN GOSUB 1370
1630 IF LEFT$(TYPE$,1)="M" OR LEFT$(TYPE$,1)="M" THEN GOSUB 1300
1640 RETURN
1650 FOR J=0 TO NF: LEO$(J)=CCUST$(J):NEXT J
1660 FOR J=NF+1 TO NF+NNUM:LEO#(J)=CVD(CCUST$(J)):NEXT J:RETURN
1670 PRINT"ORDER SELECTION MENU":TL=7
1680 GOSUB 1860
1690 IF NOFILE=-1 THEN GOSUB 1710:RETURN
1700 GOSUB 1720:RETURN
1710 FOR I=1 TO FAMOUNT:ORDER(I)=I:NEXT :RETURN
1720 GOSUB 1710
1730 Z=0
1740 Z=Z+1:GET 2,Z:GOSUB 1650:WORK$(Z)=LEO$(NOFILE)
1750 IF LEFT$(LEO$(1),3)="END" THEN Z=Z-1 ELSE GOTO 1740
1760 FOR SORT=1 TO Z
1770 FOR I=1 TO Z-SORT
1780 IF WORK$(I)<=WORK$(I+1) THEN 1830
1790 SV=ORDER(I):ORDER(I)=ORDER(I+1)
1800 ORDER(I+1)=SV
1810 SV$=WORK$(I):WORK$(I)=WORK$(I+1)
1820 WORK$(I+1)=SV$
1830 NEXT I
1840 NEXT SORT
1850 RETURN
1860 PRINT"-1","BY FILE NUMBER"
1870 FOR IX=0 TO TL:PRINT IX,LC$(IX):NEXT
1880 PRINT:INPUT NOFILE
1890 IF NOFILE<-1 OR NOFILE>TL THEN 1860
1900 RETURN
1910 PRINT:PRINT:HTAB 10:PRINT"** List Option **":PRINT:PRINT:
     HTAB 5:PRINT"Starting File No.";TAB(25);"Ending File No.":
1911 HTAB 5:PRINT"-------- ---- ---";TAB(25);"------ ---- ---":PRINT:
1912 PRINT TAB(10);:INPUT ;A:PRINT TAB(30);:INPUT B
```

```
1913 PRINT:PRINT:
1920 HOME:PRINT"NO ";:FOR J=0 TO 8 :PRINT USING ST$;LC$(J);" ";:
     NEXT :PRINT
1930 FOR JJ=A TO B:V=ORDER(JJ):GET 2,V
1940 GOSUB 1650
1950 PRINT USING"### ";V;:FOR I=0 TO NF:
     PRINT USING ST$;LEO$(I);"  ";:NEXT
1960 FOR I=NF+1 TO NF+NNUM:PRINT USING KKX$;LEO#(I);:NEXT :PRINT
1970 NEXT JJ
1980 PRINT:PRINT:HTAB 30:INPUT "<CR> TO CONTINUE";DU$
1990 GOTO 640
2000 LPRINT "NO ";:FOR J=0 TO 4:LPRINT USING KKK$;LC$(J);:
     NEXT :LPRINT
2010 JJ=0
2020 JJ=JJ+1:V=ORDER(JJ)
2030 GET 2,V
2040 GOSUB 1650
2050 IF LEFT$(LEO$(1),3)="END" THEN 2080
2060 LPRINT USING "### ";V;:FOR I=0 TO 4:LPRINT USING KKK$;LEO$(I);:
     NEXT : LPRINT
2070 GOTO 2020
2080 LPRINT
2090 LPRINT "NO ";:FOR I=5 TO 8:LPRINT USING KKK$;LC$(I);:NEXT :
     LPRINT
2100 JJ=0
2110 JJ=JJ+1:V=ORDER(JJ)
2120 GET 2,V
2130 GOSUB 1650
2140 IF LEFT$(LEO$(1),3)="END" THEN 640
2150 LPRINT USING "### ";V;:FOR I=5 TO NF:LPRINT USING KKK$;LEO$(I);:
     NEXT
2160 FOR I=NF+1 TO NF+NNUM:LPRINT USING KKX$;LEO#(I);:NEXT :LPRINT
2170 GOTO 2110
3000 REM PROMO SCREEN FOR INVOICE
3010 VTAB 3:HTAB 23:PRINT"**** MAILLIST GENERATION PROGRAM ****"
3020 VTAB 8: HTAB 39:PRINT"by":VTAB 12:HTAB 27:
3030 PRINT"Steven M. Zimmerman Ph.D.":PRINT:
3040 HTAB 33: PRINT"Leo M. Conrad":PRINT:
3050 HTAB 27:PRINT"Donald R. Smith Ph.D., P.E.":
3060 VTAB 20:HTAB 28:INPUT "( Press < CR > to Begin )";DUM$:HOME:
3065 GOSUB 3100
3070 RETURN
3100 HOME: VTAB 3:
     PRINT"* * * * *  M A I L L I S T  G E N E R A T I O N
     P R O G R A M  * * * * * ":PRINT:PRINT:
3110 PRINT"- PURPOSE OF THIS PROGRAM:":PRINT:PRINT:
3120 PRINT"This program will assist in the generation of mailing
     labels or for print-"
3130 PRINT"ing names and address on envelopes.  This program is
     designed to work with"
3140 PRINT"the database used by the check writing program and
     the invoice program."
3150 PRINT"The program requires that a file of either customers
     or suppliers be"
```

```
3160 PRINT"maintained on a diskette in drive B. Place your file
     diskette in drive"
3170 PRINT"B and press  < CR > to begin.":PRINT:PRINT:HTAB 20:
     INPUT "PRESS < CR > TO BEGIN"; DDUM$:PRINT:PRINT:
3180 RETURN
```

# Chapter 9—CHECK

```
10 PRINT:PRINT:REM: "CHECK.BAS"
15 HOME:WIDTH 80:GOSUB 2000
30 REM 3. SUPPLY.DAT RANDOM    SAVES supplier's names etc.
   trans.dat  sequential Saves or inputs transactions you name it
40 ZIP$="$$###,###,###.##":ZX$="#,###,###":
   REM if you are working with non-integer quantities change
   ZX$=... to reflect this fact
50 ZAP$="$$##,###,###,###,###.##"
60 HOME:NF=7:NNUM=1:
62 REM NF is the number of strings$ in the supplier's file
63 REM NNUM is the number of numbers--strings store first then numbers
70 LC$(0)="Supplier's name ":LC$(1)="Company's name  ":
   LC$(2)="Number and street"
80 LC$(3)="City            ":LC$(4)="State           ":
   LC$(5)="Zip code        ":LC$(6)="First name      ":
   LC$(7)="Title           "
90 LC$(8)="Balance Due    "
119 VTAB 5:HTAB 3:LINE INPUT "1. Today's Date is ";DATE$
120 PRINT:HTAB 3:INPUT"2. DISK Drive for data files (A,B,C,D)";DISK$:
    PRINT :PRINT:GOSUB 640
130 HTAB 20:PRINT"**** TRANSACTION FILE MENU ****":PRINT:HTAB 10:
    PRINT"(I)nput":HTAB 10:PRINT"(O)utput":HTAB 10:INPUT "(N)one ";NN$
132 IF LEFT$(NN$,1)="n" OR LEFT$(NN$,1)="N" THEN 160
140 PRINT:PRINT:HTAB 6:INPUT "Name of file";TRANSNAME$:
    TRANSNAME$=DISK$+":"+TRANSSAME$
150 IF LEFT$(NN$,1)="o" OR LEFT$(NN$,1)="O" THEN
    OPEN "O",#4,TRANSNAME$ ELSE OPEN "I",#4,TRANSNAME$:
    REM OPEN FOR EITHER INPUT OR OUTPUT BUT NOT BOTH
160 HOME:VTAB 2:HTAB 20:PRINT"***** FILE OUTPUT MENU *****":
    PRINT:PRINT:PRINT:
180 PRINT:PRINT"Select output type:":PRINT"     (F)ILE update "
190 PRINT"     (C)heck writing ":PRINT"     (E)ND":INPUT TYPE$
200 IF LEFT$(TYPE$,1)="E" OR LEFT$(TYPE$,1)="e" THEN
    IF LEFT$(NN$,1)="o" OR LEFT$(NN$,1)="O" THEN ENN$="END":
    PRINT #4,0;CHR$(34);ENN$;CHR$(34);CHR$(34);ENN$;CC$(34);0;0;0;0;
    0;CHR$(34);ENN$;CHR$(34))LOSE:GOTO 10
210 IF LEFT$(TYPE$,1)="F" OR LEFT$(TYPE$,1)="f" THEN 670
220 PRINT:PRINT:INPUT"New starting number (Y/N)";SNUM$
230 IF LEFT$(SNUM$,1)="n" OR LEFT$(SNUM$,1)="N" THEN
    X$=DISK$+":NOCHECK.DAT": OPEN "I",#1,X$: INPUT #1,SN,SYSTEX,
    TSYSTEM,CASH: DEBIT=SYSTEX:CLOSE 1: GOTO 250
231 REM this file contains the check no. If you are using
    preprinted numbers, this file is not needed.
240 PRINT:PRINT"Start. CK No.";TAB(17);"Accts. Pay (Low No.)";
    TAB(41);"Accts Pay (High No.)";TAB(64);"Cash Acct. NO.":
241 PRINT"------ -- ---";TAB(17);"------ --- ---- ----";
    TAB(41);"----- --- ----- ----";TAB(64);"---- ----- ---":PRINT:
242 PRINT TAB(5);:INPUT ;SN: PRINT TAB(23);:INPUT ;SYSTEX:
    PRINT TAB(46);:INPUT ;TSYSTEM:PRINT TAB(67);: INPUT CASH:
```

```
244 DEBIT=SYSTEX
250 OPEN"I",#1,"B:NAMEX.DAT":FOR I=0 TO 5:INPUT #1,COMPANY$(I):
    NEXT:CLOSE #1
260 IF LEFT$(NN$,1)="i" OR LEFT$(NN$,1)="I" THEN 1150
270 PRINT:PRINT:PRINT"You must use disk file to input to post
    transactions to individual accounts":
    INPUT "Select input method-- (K)eys or (D)isk ";IM$
272 IF LEFT$(NN$,1)="y" OR LEFT$(NN$,1)="Y" THEN
    IF LEFT$(IM$,1)="k" OR LEFT$(IM$,1)="K" THEN 270
280 IF LEFT$(IM$,1)="D" OR LEFT$(IM$,1)="d" THEN
    INPUT "Post to (I)ndividaul or (A)ccounts payable";PPOST$
290 IF LEFT$(IM$,1)="k" OR LEFT$(IM$,1)="K" THEN GOSUB 310:
    IF FXX=98 THEN 790 ELSE 380
300 IF LEFT$(IM$,1)="d" OR LEFT$(IM$,1)="D" THEN 480 ELSE 270
310 PRINT:PRINT:FOR I=0 TO NF:PRINT"Input ";LC$(I);:
    INPUT LEO$(I):NEXT I:FOR I=NF+1 TO NF+NNUM:
    PRINT"Input ";LC$(I);:INPUT LEO#(I):NEXT I
320 WIDTH 80
330 FOR I=0 TO NF:PRINT I,LC$(I),LEO$(I):NEXT I:
    FOR I=NF+1 TO NF+NNUM:PRINT I,LC$(I),LEO#(I):NEXT I
340 INPUT "99 continue, 98 search, or number to change";FXX
350 IF FXX>97 THEN RETURN
360 PRINT "New information ":IF FXX>NF THEN INPUT LEO#(FXX)
    ELSE INPUT LEO$(FXX)
370 HOME:GOTO 320
380 PRINT:PRINT:INPUT "Input invoice number ";INVNUM
390 PRINT:PRINT:INPUT"Purchase order number";PO$
400 INPUT" Amount of check ";MONEY#
410 PRINT DATE$;
420 PRINT USING ZIP$;MONEY#
430 INPUT "Input 99 if complete or new value ";CNUM#
440 IF CNUM#=99# THEN 460
450 MONEY#=CNUM#:GOTO 420
460 WIDTH 80:DAT$=DATE$:GOSUB 1240:GOTO 160
470 REM This is section II ****************************************
480 REM ouput from file
490 HOME:PRINT:PRINT:PRINT" F i l e   I n p u t   M e n u ":
    PRINT :PRINT "    -1     File number"
500 PRINT"     0     Supplier's name":
    PRINT"     1     Company's name":
    PRINT"     2     Number and street":PRINT"     3     City"
510 PRINT"     4     State":PRINT"     5     Zip code":
    PRINT"     6     First name":INPUT "     7     Title";NOFILE:
    IF NOFILE<-1 OR NOFILE>7 THEN 490
520 IF NOFILE=-1 THEN INPUT "File - Customer Number";K:GET 2,K:
    GOTO 590
530 INPUT "Name to be searched for";SEARCH$
540 K=0
550 K=K+1:KK=LEN(SEARCH$):IF NOFILE=-1 THEN GET 2,K:GOTO 590
560 GET 2,K:K$=CCUST$(NOFILE):IF LEFT$(K$,KK)=SEARCH$ THEN 590
570 IF LEFT$(K$,3)="END" THEN
    INPUT "Cannot find file Input to try again";DU$:
    HOME:CLOSE 2:GOTO 160
580 GOTO 550
```

```
590 FOR J=0 TO NF:LEO$(J)=CCUST$(J):NEXT J:FOR J=NF+1 TO NF+NNUM:
    LEO#(J)=CVD(CCUST$(J)):NEXT J
600 GOSUB 320:IF FXX=98 THEN 550
610 GOTO 380
620 REM This is section III ************************************
630 REM lines are to open customer.data file
640 X$=DISK$+":SUPPLY.DAT"
650 OPEN "R",2,X$,111:FIELD #2,12 AS CCUST$(0),25 AS CCUST$(1),
    25 AS CCUST$(2),15 AS CCUST$(3),3 AS CCUST$(4),5 AS CCUST$(5),
    8 AS CCUST$(6),10 AS CCUST$(7),8 AS CCUST$(8):RETURN
660 REM SUPPLIERS NAME (12) CO NAME (25) NO AND STREET (25) CITY (15)
662 REM STATE (3) ZIP (5) FIRST NAME (8) TITLE (10) REM BAL (8)
670 REM lines 700 through 950 are for file mainentance
680 HOME:K$="":PRINT:PRINT:
    PRINT" F i l e   M a i n t e n a n c e   M e n u "
690 PRINT:PRINT"     (A)dd new supplier":
    PRINT"     (D)ELETE a file":PRINT "     (E)nd cycle":
    PRINT "     (S)TART new file"
700 PRINT"     (U)pdate an existing file":INPUT" ";FILE$
710 IF LEFT$(FILE$,1)="E" OR LEFT$(FILE$,1)="e" THEN HOME:GOTO 160
720 IF LEFT$(FILE$,1)="A" OR LEFT$(FILE$,1)="a" THEN GOSUB 310:
    GOTO 770
730 IF LEFT$(FILE$,1)="D" OR LEFT$(FILE$,1)="d" THEN 820
740 IF LEFT$(FILE$,1)="S" OR LEFT$(FILE$,1)="s" THEN 990
750 IF LEFT$(FILE$,1)="U" OR LEFT$(FILE$,1)="u" THEN 820
760 GOTO 680
770 K=0
780 K=K+1
790 GET 2,K:K$=CCUST$(0):IF LEFT$(K$,3)="END" OR LEFT$(K$,3)="OUT"
    THEN 800 ELSE 780
800 FOR J=O TO NF:LSET CCUST$(J)=LEO$(J):NEXT J
802 FOR J = NF+1 TO NF+NNUM:LSET CCUST$(J)=MKD$(LEO#(J)):NEXT J:
804 PUT 2,K: IF LEFT$(K$,3)="END" THEN FOR J=0 TO NF:
    LSET CCUST$(J)="END":NEXT J
806 FOR J =NF+1 TO NF +NNUM:LSET CCUST$(J)=MKD$(9999999999#):NEXT J:
    PUT 2,K+1
810 GOTO 680
820 PRINT :PRINT :PRINT "S e a r c h   M e n u":
    PRINT "    -1     By file number"
830 PRINT"     0     Supplier's name":
    PRINT"     1     Company's name":
    PRINT"     2     Number and street":PRINT"     3     City"
840 PRINT"     4     State":PRINT"     5     Zip Code":
    PRINT"     6     First name":PRINT"     7     Title":
    INPUT "     8     Current balance ";NOFILE: IF NOFILE >8 THEN 820
850 IF NOFILE=-1 THEN INPUT "File - Supplier's Number";K:
    GET 2,K:GOTO 930
860 IF NOFILE=8 THEN 1020
870 INPUT "Name to be searched for";SEARCH$
880 K=0
890 K=K+1:KK=LEN(SEARCH$):IF NOFILE=-1 THEN GET 2,K:GOTO 930
900 GET 2,K:K$=CCUST$(NOFILE):PRINT K$:
    IF LEFT$(K$,KK)=SEARCH$ THEN 930
910 IF LEFT$(K$,3)="END" THEN
    INPUT "Cannot find file Input to try again";DU$:GOTO 680
```

```
920 GOTO 890
930 FOR J=0 TO NF:LEO$(J)=CCUST$(J):NEXT J:FOR J=NF+1 TO NF+NNUM:
    LEO#(J)=CVD(CCUST$(J)):NEXT J
940 GOSUB 320
950 IF FXX=98 THEN 890
960 IF LEFT$(FILE$,1)="d" OR LEFT$(FILE$,1)="D" THEN 980
970 GOTO 800
980 FOR J=0 TO NF:LSET CCUST$(J)="OUT":NEXT J:
    FOR J=NF+1 TO NF+NNUM:LSET CCUST$(J)=MKD$(9999999999#):NEXT J:
    PUT 2,K:GOTO 680
990 WIDTH 80:PRINT:PRINT:PRINT:
    INPUT "Are you sure you wish to continue? Type completely..
    YES or NO";YY$:IF YY$<>"YES" THEN GOTO 160
1000 FOR I=0 TO NF:LSET CCUST$(I)="END":NEXT I:
     FOR J=NF+1 TO NF+NNUM:LSET CCUST$(J)=MKD$(9999999999#):
     NEXT J:PUT 2,1:PUT 2,2
1010 WIDTH (80):GOTO 680
1020 HOME:PRINT "Value search alternatives":PRINT "     EQUAL":
     PRINT "     GREATER":PRINT "     LESS";:INPUT VALUE$
1030 INPUT "Value ";VALUE#:IF VALUE# <0 THEN 1020
1040 K=0
1050 K=K+1:GET 2,K:DD#=CVD(CCUST$(NOFILE)):
     IF LEFT$(VALUE$,1)="E" OR LEFT$(VALUE$,1)="e" THEN
     IF DD#=VALUE# THEN 1100
1060 IF LEFT$(VALUE$,1)="G" OR LEFT$(VALUE$,1)="g" THEN
     IF DD#>VALUE# THEN 1100
1070 IF LEFT$(VALUE$,1)="L" OR LEFT$(VALUE$,1)="l" THEN
     IF DD#<VALUE# THEN 1100
1080 IF DD#<99999999# THEN 1050
1090 INPUT "Cannot find file Input to try again";DU$:GOTO 680
1100 FOR J=0 TO NF:LEO$(J)=CCUST$(J):NEXT J:FOR J=NF+1 TO NF+NNUM:
     LEO#(J)=CVD(CCUST$(J)):NEXT J
1110 GOSUB 320
1120 IF FXX=98 THEN 1050
1130 GOTO 680
1140 REM Routine to read in transaction file from which checks printed
1150 INPUT #4,NUMBER,DAT$,DESCRIP$,K,AC,AD#,MONEY#,STATE$:INVNUM=NUMBER
1160 IF DAT$="END" THEN CLOSE 4:GOTO 160
1170 J%=K:
     K=K-SYSTEX:
     REM must work from individual accounts when automaticly writing
     checks
1180 IF J%<SYSTEX OR J%>TSYSTEM THEN
     INPUT "The account numbers are beyond the acceptable range enter
     <CR> to restart system";DU$:CLOSE:GOTO 10
1190 GET 2,K
1200 FOR J=0 TO NF:LEO$(J)=CCUST$(J):NEXT J
1210 FOR J=NF+1 TO NF+NNUM:LEO(J)=CVD(CCUST$(J)):NEXT J
1220 GOSUB 1240:GOTO 1150
1230 REM This is section IV ****************************************
1240 LEFT=10
1250 LPRINT:LPRINT:LPRINT
1260 REM replace the next 6 statements with 8 LPRINT's if checks have
     printed name
1270 LPRINT TAB(LEFT+3)COMPANY$(0)
```

```
1280 LPRINT TAB(LEFT+3)"Accounts Payable"
1290 LPRINT TAB(LEFT+3)COMPANY$(1)
1300 LPRINT TAB(LEFT+3)COMPANY$(2);",";COMPANY$(3);" ";COMPANY$(4):
     LPRINT
1310 LPRINT TAB(LEFT+3)COMPANY$(5)
1320 LPRINT TAB(LEFT+24);USING "\         \";DAT$;:
     LPRINT USING ZIP$;MONEY#
1330 LPRINT :LPRINT
1340 FOR J=0 TO 3:LPRINT TAB(LEFT+3) LEO$(J);:NEXT J
1350 LPRINT ",  ";LEO$(4);"  ";LEO$(5):LPRINT
1360 LPRINT:LPRINT :LPRINT
1370 SN=SN+1:X$=DISK$+":NOCHECK.DAT":OPEN "O",#1,X$:
     PRINT #1,SN,SYSTEX,TSYSTEM,CASH:CLOSE 1:
     REM file contains only the check no. May drop if preprinted
     numbers used.
1380 LPRINT
1390 LPRINT TAB(LEFT+3)COMPANY$(0):REM or LPRINT if name printed
     on checks
1400 LPRINT
1410 LPRINT "Supplier's No.","Invoice Date","Invoice Amount":LPRINT
1420 LPRINT INVNUM,,DAT$,USING "$$####,###,###.##";MONEY#
1430 NVV=15
1440 FOR I=1 TO NVV:LPRINT :NEXT I
1450 IF LEFT$(NN$,1)="o" OR LEFT$(NN$,1)="O" THEN GOSUB 1470
1460 RETURN
1470 REM this subroutine posts the information to the tranaction
     file as defined and to the supplier's file
1480 IF LEFT$(IM$,1)="K" OR LEFT$(IM$,1)="k" THEN K=0
1490 IF LEFT$(PPOST$,1)="i" OR LEFT$(PPOST$,1)="I" THEN
     J%=K+SYSTEX ELSE J%=SYSTEX
1500 SUM#=MONEY#:DOLLAR#=SUM#:GOSUB 1600:SUM#=DOLLAR#
1502 IF J%<SYSTEX OR J%>TSYSTEM THEN
     PRINT"Your account number is beyond the limit allowed by your
     accounting system ";J%;" is greater than ";TSYSTEM-SYSTEX;:
     INPUT  " or less than 0-<CR> TO CONTINUE";D
1510 Y$="ISSUED":PRINT #4,SN;CHR$(34);DATE$;CHR$(34);CHR$(34);
     LEO$(1);CHR$(34);J%%ASH;SUM#;SUM#;CHR$(34);;CHR$(34)
1520 IF LEFT$(IM$,1)="k" OR LEFT$(IM$,1)="K" THEN RETURN
1530 IF LEFT$(PPOST$,1)="a" OR LEFT$(PPOST$,1)="A" THEN RETURN
1540 SUM#=-SUM#+CVD(CCUST$(8))
1550 FOR J=0 TO NF-1:LSET CCUST$(J)=LEO$(J)
1560 NEXT J:LSET CCUST$(8)=MKD$(SUM#):PUT 2,K
1570 RETURN
1580 REM Have to run transaction file through an account payable
     journal to change k and to post to suppler's file
1590 REM the transaction file includes trans# date$ description$
     acct db/cr, amt db/cr, and clear$
1600 XXXX#=INT(DOLLAR#*100#)/100#:DIFF#=DOLLAR#-XXXX#:
     IF DIFF#>5.000000000000007D-03 THEN XXXX#=XXXX#+.01#
1610 DOLLAR#=XXXX#:RETURN:REM this subroutine corrects for
     round-off errors
2000 REM PROMO SCREEN FOR INVOICE
2010 VTAB 3:HTAB 23:PRINT"****CHECK WRITING PROGRAM ****"
2020 VTAB 8: HTAB 39:PRINT"by":VTAB 12:HTAB 27:
2030 PRINT"Steven M. Zimmerman Ph.D.":PRINT:
```

```
2040 HTAB 33: PRINT"Leo M. Conrad":PRINT:
2050 HTAB 27:PRINT"Donald R. Smith Ph.D., P.E.":
2060 VTAB 20:HTAB 28:INPUT "( Press < CR > to Begin )";DUM$:HOME:
2065 GOSUB 2100
2070 RETURN
2100 REM PGM DIRECTIONS SCREEN
2110 HOME:VTAB 3:
2120 PRINT"The checkwriting program is designed to produce checks
     from a data file"
2130 PRINT"of transactions and suppliers or, to work directly from
     the keyboard"
2140 PRINT"without a file of suppliers.  The checks produced by the
     program re-"
2150 PRINT"quire that custom checks used with a tractor-feed printer
     be purchased"
2160 PRINT"and utilized.  Your printer must have tractor-feed
     capability."
2170 PRINT:PRINT"If you are going to use this feature, install
     the checks in your"
2180 PRINT"printer and align to top-of-form":VTAB 20:
     HTAB 25:INPUT "Press <CR> to Continue";DDUM$:HOME: VTAB 5:
2190 RETURN


^@
```

# Chapter 11—EVALPAC.BAS

```
10 REM "EVALPAC"
20 HOME:CLEAR 500:GOSUB 4000:GOTO 70
30 DIM DE(100),AL(100),TALK$(12),A(16)
32 DIM VIDEO%(80,21)
35 EIGHT=80
40 A$="###,###,###,###.##":RETURN
70 GOSUB 30
80 PRINT:GOTO 100
90 CLEAR:HOME:PRINT:PRINT "EVALPAC":GOSUB 30
100 WIDTH 80
105 VTAB 3:HTAB 25:PRINT"***  EVALPAC MAIN MENU  ***":PRINT:PRINT:
110 PRINT "                (A)MORTIZATION OF LOAN"
120 PRINT "                (B)REAK-EVEN ANALYSIS"
130 PRINT "                (C)ALL -BALLOON NOTE VALUE"
140 PRINT "                (D)ECLINING BALANCE DEPRECIATION"
150 PRINT "                (E)ND"
160 PRINT "                (F)UTURE VALUE OF AN ASSET"
170 PRINT "                (M)ORTGAGE CALCULATIONS"
190 PRINT "                (P)RESENT VALUE OF UNEQUAL CASH FLOWS"
210 PRINT "                (R)EAL ESTATE PROPERTY 1981 LAW
                           DEPRECIATION"
220 PRINT "                (S)TRAIGHT LINE DEPRECIATION"
230 PRINT "                (V)ALUE -DISCOUNTED OF A BOND/MORTGAGE"
240 PRINT "                (X)DEPRECIATION PERSONAL PROPERTY
                           1982 CHANGES "
250 PRINT "                (Y)EARS DIGITS -SUM OF  -- DEPRECIATION"
252 PRINT "                (*)CASH FLOW ANALYSIS FOR REAL
                           ESTATE INVESTMENT"
254 PRINT "                (#)FORECAST Regression and
                           Correlation Analysis"
260 PRINT:PRINT:HTAB 16:INPUT "*** Enter Your Choice- ";SL$:WIDTH 80
270 IF LEFT$(SL$,1)="E" OR LEFT$(SL$,1)="E" THEN END
290 IF LEFT$(SL$,1)="M" OR LEFT$(SL$,1)="M" THEN 430
300 IF LEFT$(SL$,1)="A" OR LEFT$(SL$,1)="A" THEN 800
320 IF LEFT$(SL$,1)="C" OR LEFT$(SL$,1)="C" THEN
    SM$="C":SL$="C":HOME:GOTO 520
330 IF LEFT$(SL$,1)="F" OR LEFT$(SL$,1)="F" THEN 1460
340 IF LEFT$(SL$,1)="P" OR LEFT$(SL$,1)="P" THEN 1540
350 IF LEFT$(SL$,1)="S" OR LEFT$(SL$,1)="S" THEN HOME::GOTO 1660
360 IF LEFT$(SL$,1)="Y" OR LEFT$(SL$,1)="Y" THEN HOME:GOTO 1660
370 IF LEFT$(SL$,1)="X" OR LEFT$(SL$,1)="X" THEN 2380
380 IF LEFT$(SL$,1)="R" OR LEFT$(SL$,1)="R" THEN 2770
390 IF LEFT$(SL$,1)="V" OR LEFT$(SL$,1)="V" THEN 3250
400 IF LEFT$(SL$,1)="B" OR LEFT$(SL$,1)="B" THEN 3420
410 IF LEFT$(SL$,1)="D" OR LEFT$(SL$,1)="D" THEN HOME:
    PRINT "DECLINING BALANCE DEPRECIATION":GOTO 1660
412 IF LEFT$(SL$,1)="*" THEN RUN "CASH"
414 IF LEFT$(SL$,1)="#" THEN RUN "FORECAST"
```

```
420 GOTO 90
430 REM FOR MORTGAGE CALCULATIONS
440 HOME:VTAB 3:HTAB 15:
    PRINT"***** MORTGAGE CALCULATION PROGRAM *****":VTAB 6:
    PRINT "MORTGAGE MENU Solve for one of the following:":PRINT:PRINT:
450 PRINT "     (P)AYMENT":PRINT "     (I)NVESTMENT"
460 PRINT "     (A)PR":PRINT "     (L)IFE":PRINT:
    INPUT "Enter your choice: ";SM$:PRINT:
470 IF LEFT$(SM$,1)="P" OR LEFT$(SM$,1)="P" THEN SM$="PAYMENT"
480 IF LEFT$(SM$,1)="I" OR LEFT$(SM$,1)="I" THEN SM$="PRESENT VALUE"
490 IF LEFT$(SM$,1)="A" OR LEFT$(SM$,1)="A" THEN SM$="APR"
500 IF LEFT$(SM$,1)="L" OR LEFT$(SM$,1)="L" THEN SM$="LIFE"
510 IF SM$<>"PAYMENT" THEN INPUT "PAYMENT";PAY#
520 HOME:HTAB 15:PRINT"***** CALL BALLOON NOTE VALUE PROGRAM *****":
    VTAB 3:IF SM$<>"PRESENT VALUE" THEN
    INPUT "PRESENT VALUE - INVESTMENT";PV#
530 IF SM$<>"APR" THEN INPUT "APR";APR#
540 IF SM$<>"LIFE" THEN INPUT "LIFE IN YEARS";YR#
550 INPUT "PAYMENTS PER YEAR";PY#:MO#=PY#*YR#:IN#=APR#/(100#*PY#)
560 IF SM$="C" THEN 580
570 IF SM$<>"PAYMENT" THEN 610
580 PAY#=PV# * IN# * (1+IN#)^MO#/((1+IN#)^MO#-1)
590 IF SM$="C" THEN IF SL$="C" THEN 1410 ELSE 3360
600 GOTO 730
610 IF SM$<>"PRESENT VALUE" THEN 640
620 PV#=PAY# * ((1+IN#)^MO#-1)/(IN# * (1+IN#)^MO#)
630 GOTO 730
640 IF SM$<>"APR" THEN 710 ELSE PRINT
    "LINEAR APPROXIMATION USED FOR       APR ":
645 INPUT "<CR> TO CONTINUE";DU$:BOT#=.25/PY#:TP#=.75/PY#
650 FOR U=1 TO 5:C#=BOT#:D#=PAY#-PV#*C#*(1+C#)^MO# / ((1+C#)^MO# -1)
660 E#=TP#:F#=PAY#-PV#*E#*(1+E#)^MO# /((1+E#)^MO# -1)
670 A#=(D#-F#)/(C#-E#):B#=D#-A#*C#:IN#=-B#/A#
680 G#=PV#*IN#*(1+IN#)^MO# /((1+IN#)^MO# -1):
    TP#=IN#+.2/U^4:BOT#=IN#-.2/U^4
690 IF BOT#<0 THEN BOT#=.001
700 NEXT U:GOTO 730
710 IF SM$<>"LIFE" THEN 440
720 MO#=LOG(1/(1-(IN# * PV# / PAY#)))/LOG(1+IN#)
730 PRINT:PRINT:PRINT "PAYMENT:";USING A$;PAY#
740 PRINT "LOAN:";USING A$;PV#
750 PRINT "LIFE:";MO#/PY#:PRINT "PAYMENTS/YEAR:";PY#
760 PRINT PY#*100*IN#;"% "
770 IF SM$="C" THEN PRINT "BALLOON NOTE";USING A$;BV#:
    PRINT "BALLOON PAYMENT #";NO%:
775 IF LEFT$(SL$,1)="V" OR LEFT$(SL$,1)="V" THEN 3380
780 VTAB 20:HTAB 25:INPUT"<CR> TO CONTINUE";DU$:GOSUB 3880
790 GOTO 90
800 REM FOR AMORTIZATION OF A LOAN
810 A$="### ###,###,###.## ###,###,###.## ###,###,
    ###.## ###,###,###.##"
820 HOME: VTAB 3:HTAB 30:PRINT"***  OUTPUT MENU ***":
    PRINT:PRINT:HTAB 10:PRINT"(C)RT":HTAB 10:
    PRINT"(P)rinter":PRINT:HTAB 10:INPUT "Enter Your Choice C/P ";P$
830 IF LEFT$(P$,1)="C" OR LEFT$(P$,1)="C" THEN P$="C":GOTO 860
```

```
840 IF LEFT$(P$,1)="P" OR LEFT$(P$,1)="P" THEN P$="P":GOTO 860
850 GOTO 820
860 B$="LOAN, APR, PAYMENTS PER YEAR,
    PAYMENT              (0 TO CALCULATE), YEARS":
862 B1$="LOAN, APR, PAYMENTS PER YEAR, REDUCTION, YEARS":
864 B2$="LOAN, APR, PAYMENTS PER YEAR, YEARS"
865 IF EIGHT=80 THEN B$="LOAN, APR, PAYMENTS PER YEAR,
    PAYMENT (0 TO CALCULATE),YEARS"
870 PRINT:PRINT:INPUT "BALLOON (Y/N)";BL$:
    IF LEFT$(BL$,1)="Y" OR LEFT$(BL$,1)="Y" THEN
    INPUT "PAYMENT # OF BALLOON";BB%:BL$="Y"
880 INPUT "MONTH NUMBER IN WHICH PAYMENT STARTS";M%
890 PRINT:PRINT:PRINT "PAYOUT MENU":PRINT" EQUAL (P)AYMENT":
    PRINT" EQUAL (R)EDUCTION":INPUT " (I)NTEREST ONLY ";T$
900 IF LEFT$(T$,1)="P" OR LEFT$(T$,1)="P" THEN T$="P"
910 IF T$<>"P" THEN 1050
920 GOSUB 4100:APR#=.01*APR#:IF PAY#<>0 THEN 950
930 R#=APR#/PP%: II#=(1#+R#)^(PP%*YEAR):
    FF#=(R#*II#)/(II#-1):PAY#=FF#*LO#
940 D#=100*PAY#-INT(100*PAY#):IF D#>.5 THEN
    PAY#=PAY#-D#/100.001953125#+.01#  ELSE PAY#=PAY#-D#/100#
950 GOSUB 1250
960 R#=APR#/PP%:N%=PP%*YEAR:IF BL$="Y" THEN LET N%=BB%
970 IS#=0:RS#=0:K%=M%-2
980 FOR B=1 TO N%-1:K%=K%+1:IF K%=PP% AND PP%>1 THEN K%=0:GOSUB 1350
990 IN#=R#*LO#:D#=100*IN#-INT(100*IN#):
    IF D#>.5 THEN IN#=IN#-D#/100!+.01# ELSE IN#=IN#-D#/100#
1000 RE#=PAY#-IN#: LO#=LO#-RE#: IS#=IS#+IN#:RS#=RS#+RE#:C=B:GOSUB 1390
1010 NEXT
1020 C=N%:IN#=R#*LO#: RE#=LO#:LO#=0!:
     IS#=IS#+IN#:RS#=RS#+RE#:GOSUB 1390
1030 GOSUB 1350
1040 GOTO 90
1050 IF LEFT$(T$,1)="R" OR LEFT$(T$,1)="R" THEN T$="R"
1060 IF T$<>"R" THEN 1150
1070 PAY#=0:GOSUB 4100:APR#=.01*APR#:GOSUB 1250
1080 R#=APR#/PP%:N%=PP%*YEAR:IF BL$="Y" THEN LET N%=BB%
1090 IS#=0:RS#=0:K%=M%-2:SS#=RE#
1100 FOR B=1 TO N%-1:K%=K%+1:IF K%=PP% AND PP%>1 THEN K%=0:GOSUB 1350
1110 RE#=SS#:IN#=R#*LO#:D#=100*IN#-INT(100*IN#):
     IF D#>.5# THEN IN#=IN#-D#/100#+.01# ELSE IN#=IN#-D#/100#
1120 LO#=LO#-RE#: IS#=IS#+IN#:RS#=RS#+RE#:C=B:GOSUB 1390
1130 NEXT
1140 GOTO 1020
1150 IF LEFT$(T$,1)="I" OR LEFT$(T$,1)="I" THEN T$="I"
1160 IF T$<>"I" THEN 890
1170 PRINT B2$:INPUT LO#,APR#,PP%,YEAR:APR#=.01#*APR#:
     RE#=0:PAY#=0:GOSUB 1250
1180 R#=APR#/PP%:N%=PP%*YEAR:IF BL$="Y" THEN N%=BB%
1190 IS#=0:RS#=0:K%=M%-2:SS#=0
1200 FOR B=1 TO N%-1:K%=K%+1:IF K%=PP% AND PP%>1 THEN K%=0:GOSUB 1350
1210 RE#=SS#:IN#=R#*LO#:D#=100#*IN#-INT(100#*IN#):
     IF D# >.5#  THEN IN#=IN#-D#/100#+.01# ELSE IN#=IN#-D#/100#
1220 IS#=IS#+IN#:C=B:GOSUB 1390
1230 NEXT
```

```
1240 GOTO 1020
1250 IF P$="C" THEN 1300
1260 INPUT "TITLE, DATE (XX/XX/XX)";TT$,D$:LPRINT TT$,D$
1270 LPRINT "LOAN= ";LO#,"APR= ";APR#,"PAYMENTS PER YEAR= ";PP%
1280 LPRINT "YEARS= ",YEAR,"PERIOD PAYMENT= ";PAY#
1290 LPRINT "PERIOD       INTEREST     REDUCTION
     LOAN          PAYMENT"
1300 PRINT "LOAN= ";LO#,"APR= ";APR#,"PAYMENTS PER YEAR= ";PP%
1310 PRINT "YEARS= ";YEAR,"PERIOD PAYMENT= ";PAY#
1320 WIDTH 80:PRINT "PERIOD        INTEREST     REDUCTION
           LOAN          PAYMENT"
1330 GOSUB 1380
1340 RETURN
1350 C=0:IN#=IS#:IS#=0:RE#=RS#:RS#=0:GOSUB 1390
1360 IF P$<>"P" THEN GOSUB 1380
1370 RETURN
1380 INPUT "<CR> TO PAGE";DU$:RETURN
1390 PRINT USING A$;C,IN#,RE#,LO#,IN#+RE#:IF P$<>"C" THEN
     LPRINT USING A$;C,IN#,RE#,LO#,IN#+RE#
1400 RETURN
1410 REM FOR BALLOON NOTE CALCULATIONS
1420 INPUT "PAYMENT # OF BALLOON PAYMENT";NO%
1430 LF#=MO#-NO%
1440 BV#=PAY# * ((1+IN#)^LF#-1)/(IN# * (1+IN#)^LF#)
1450 GOTO 730
1460 REM FUTURE VALUE OF CURRENT VALUE
1470 HOME:VTAB 4:HTAB 15:
     PRINT"*****  FUTURE VALUE OF A PRESENT SUM OF MONEY *****":
     PRINT:PRINT:PRINT:INPUT "PRESENT VALUE";PV#
1480 INPUT "INFLATION OR EARNING RATE";IN#:IN#=.01*IN#
1490 INPUT "NUMBER OF PERIODS";NO%
1500 FF#=PV#*(1+IN#)^NO%
1510 PRINT "FUTURE VALUE:";FF#
1520 VTAB 20:HTAB 32::INPUT "<CR> TO CONTINUE";DU$:GOSUB 3880
1530 GOTO 90
1540 REM PRESENT VALUE OF UNEQUAL CASH FLOWS
1550 HOME:SUM#=0!:VTAB 3:HTAB 15:
     PRINT"***** PRESENT VALUE OF UNEQUAL CASH FLOWS ****":
     VTAB 6:INPUT"DISCOUNT RATE (%)";IN#:IN#=.01*IN#
1560 INPUT "(C)RT OR (P)RINTER ";YN$
1570 IF LEFT$(YN$,1)="P" OR LEFT$(YN$,1)="p" THEN YN$="P" :GOTO 1575
1572 GOTO 1580
1575 LPRINT "DISCOUNT RATE ";100*IN#:
     LPRINT "YEAR","VALUE","SUM OF DISCOUNTED VALUES"
1580 PRINT:PRINT:HTAB 15:PRINT"YEAR";TAB(30);"CASH FLOW AMOUNT ($)":
     HTAB 15:PRINT"----";TAB(30);"--------------------";SPC(2);
     "Enter 0,0 to END":PRINT:PRINT:
1582 PRINT TAB(16);:INPUT ;NO%:PRINT TAB(32);:INPUT ;PV#
1590 IF PV#=0 AND NO%=0 THEN 1630
1600 SUM#=SUM#+PV#/(1+IN#)^NO%
1610 IF YN$="P" THEN LPRINT NO%,PV#,SUM#
1620 GOTO 1582
1630 PRINT:PRINT:PRINT "PRESENT VALUE: $";SUM#
1640 PRINT:PRINT:PRINT:HTAB 25:INPUT "<CR> TO CONTINUE";DU$
1650 GOTO 90
```

```
1660 REM DECLINING BAL DEPRECIATION
1661 IF LEFT$(SL$,1)="S" OR LEFT$(SL$,1)="s" THEN
     TAG$="***** STRAIGHT LINE DEPRECIATION *****":GOTO 1665
1662 IF LEFT$(SL$,1)="Y" OR LEFT$(SL$,1)="y" THEN
     TAG$="***** SUM OF THE YEARS DIGITS DEPRECIATION"::GOTO 1665
1663 TAG$="***** DECLINING BALANCE DEPRECIATION *****"
1665 HOME:VTAB 3:HTAB 20:PRINT TAG$:VTAB 6:
1670 INPUT "(C)RT, (P)RINTER ";DU$
1680 TALK$(1)="YEAR OF PURCHASE"
1690 TALK$(2)="DEPRECIABLE BASIS"
1700 TALK$(3)="SALVAGE VALUE"
1710 TALK$(4)="LIFE"
1720 TALK$(5)=SL$:D(5)=0
1730 TALK$(6)="MONTH OF PURCHASE"
1740 TALK$(7)="RATE"
1750 TALK$(8)= "DEPRECIATION"
1760 FOR I=1 TO 4
1770 PRINT TALK$(I);
1780 INPUT D(I):NEXT:PRINT TALK$(6);:INPUT D(6)
1790 IF LEFT$(SL$,1)="D" OR LEFT$(SL$,1)="D" THEN
     PRINT TALK$(7):INPUT D(7)
1800 S1=D(1)
1810 S2=D(2)
1820 D(8)=2
1830 IF LEFT$(DU$,1)="P" OR LEFT$(DU$,1)="p" THEN D(8)=1
1840 TALK$(9)="CUMULATIVE": TALK$(10)="BOOK VALUE"
1850 IF D(8)=1 THEN TALK$(11)="ASSET IDENTIFICATION":
     LPRINT" ": TALK$(12)="DATE :":PRINT TALK$(11):
     INPUT T$:PRINT TALK$(12)," (XX/XX/XX)":INPUT DD$:
     LPRINT TALK$(11),T$:LPRINT " ":LPRINT TALK$(12),DD$
1860 IF D(8)=1 THEN FOR J=1 TO 7:LPRINT TALK$(J),D(J):NEXT J
1870 WT$="\  \      \          \   \        \    \         \"
1880 SU=0
1890 I=1
1900 WX$="   ####  #########.## #########.## #########.##"
1910 PRINT USING WT$;TALK$(1),TALK$(8),TALK$(9),TALK$(10)
1920 IF D(8)=1 THEN LPRINT USING WT$;TALK$(1),TALK$(8),TALK$(9),
     TALK$(10)
1930 IF LEFT$(SL$,1)="S" OR LEFT$(SL$,1)="S" THEN 2190
1940 IF LEFT$(SL$,1)="Y" OR LEFT$(SL$,1)="Y" THEN 2090
1950 DR=1/D(4)
1960 DR=DR*D(7)
1970 I=0
1980 I=I+1
1990 DX=INT(100*DR*D(2)+.5)/100
2000 SX=INT(100*((D(2)-D(3))/(D(4)-I+1))+.5)/100:
2002 IF D(3)>(D(2)-DX) THEN DX=INT(100*(D(2)-D(3))+.5)/100:
2004 SU=SU+DX:D(2)=D(2)-DX:DE(I)=DX:K%=I:
2006 GOSUB 2330:D(1)=D(1)+1:IF I<D(4) THEN 1980 ELSE GOTO 2220
2010 IF SX>DX THEN 2190
2020 SU=SU+DX
2030 D(2)=D(2)-DX
2040 DE(I)=DX
2050 K%=I:GOSUB 2330
2060 D(1)=D(1)+1
```

```
2070 IF I<D(4) THEN 1980
2080 GOTO 2190
2090 SD=(D(4)*(D(4)+1))/2
2100 BV=D(2)
2110 FOR I=1 TO D(4)
2120 DX=INT(100*(((D(4)-(I-1))/SD)*(BV-D(3)))+.5)/100
2130 SU=SU+DX
2140 D(2)=D(2)-DX
2150 K%=I
2160 DE(I)=DX
2170 GOSUB 2330
2180 D(1)=D(1)+1:NEXT I:GOTO 2220
2190 DX=INT(100*((D(2)-D(3))/(D(4)-I+1))+.5)/100
2200 FOR J=I TO D(4): SU=SU+DX:D(2)=D(2)-DX: DE(J)=DX:K%=J:GOSUB 2330
2210 D(1)=D(1)+1:NEXT J
2220 INPUT "DO YOU WISH MONTHLY ALLOCATION (Y OR N)";Y$:
     IF LEFT$(Y$,1)="N" OR LEFT$(Y$,1)="N" THEN 90 ELSE Y$="Y"
2230 IF Y$<>"Y" THEN 2220
2240 IF D(8)=1 THEN LPRINT" ":LPRINT "MONTHLY ALLOCATION ":
     LPRINT" ":LPRINT USING WT$;TALK$(1),TALK$(8),TALK$(9),TALK$(10)
2250 P=(13-D(6))/12:P2=1-P:AL(1)=P*DE(1)
2260 FOR I=1 TO D(4): AL(I+1)=INT(100*(DE(I)*P2+DE(I+1)*P)+.5)/100 :
     NEXT I:D(1)=S1:D(2)=S2:SU=0
2270 FOR I=1 TO D(4)+1:DX=AL(I):SU=SU+AL(I):
     D(2)=INT(100*(D(2)-AL(I))+.5)/100
2280 K%=I
2290 GOSUB 2330
2300 D(1)=D(1)+1
2310 NEXT I:INPUT "<CR> TO CONTINUE";DU$
2320 GOTO 90
2330 PRINT USING WX$;D(1),DX,SU,D(2):X%=INT(K%/21)
2340 X%=INT(X%*21)
2350 IF K%=X% THEN INPUT"<CR> TO CONTINUE";G$:K%=0
2360 IF D(8)=1 THEN LPRINT USING WX$;D(1),DX,SU,D(2)
2370 RETURN
2380 WIDTH 80:REM PERSONAL PROPERTY DEPRECIATION
2390 HOME:VTAB 3:HTAB 5:
     PRINT"1981 DEPRECIATION RULES-PERSONAL PROPERTY":
     VTAB 6:GOTO 2460
2400 HOME:PRINT"YEAR                DEPRECIATION
                    BOOK VALUE"
2410 IF LEFT$(P$,1)="P" OR LEFT$(P$,1)="P" THEN
     INPUT"TITLE, DATE, OPERATOR";T$,D$,O$:
2412 LPRINT " ":LPRINT T$:LPRINT D$:LPRINT O$:LPRINT" ":
     LPRINT "COST:$";FC:
2414 LPRINT "YEAR                    DEPRECIATION
               BOOK VALUE"
2420 Z$="####         ###,###,###,###.##     ###,###,###,###,###.##":
     RETURN
2430 X=INT(100*(.01*A(I)*FC)+.5)/100:Y=X+Y:PRINT USING Z$;YR;X;FC-Y
2440 IF LEFT$(P$,1)="P" OR LEFT$(P$,1)="P" THEN
     LPRINT USING Z$;YR;X;FC-Y
2450 RETURN
2460 INPUT "(C)RT  (P)RINTER ";P$
2470 PRINT:INPUT "DEPRECIABLE BASIS ($)= ";FC
```

```
2480 INPUT"YEAR (19XX) ";YR:IF YR<1981 THEN 2480
2490 PRINT"LIFE OF ASSET":PRINT" 3   5   10   15 "
2500 INPUT L
2510 IF L=3 THEN 2560
2520 IF L=5 THEN 2560
2530 IF L=10 THEN 2560
2540 IF L=15 THEN 2560
2550 GOTO 2490
2560 RESTORE:GOSUB 2400:
     REM NOT USED DUE TO 1982 LAW CHANGE
     IF YR=>1985 THEN FOR I=1 TO 33:READ V:NEXT I:
     IF YR >1985 THEN FOR I=1 TO 33:READ V:NEXT I
2570 FOR I=1 TO 3:READ A(I):NEXT:IF L>3 THEN 2600
2580 RESTORE:Y=0:FOR I=1 TO L:GOSUB 2430:YR=YR+1
2590 NEXT I:PRINT:HTAB 32:INPUT "<CR> TO CONTINUE";D$:GOTO 90
2600 FOR I=1 TO 5:READ A(I):NEXT:IF L>5 THEN 2620
2610 GOTO 2580
2620 FOR I=1 TO 10:READ A(I):NEXT:IF L>11 THEN 2640
2630 GOTO 2580
2640 FOR I=1 TO 15:READ A(I):NEXT:GOTO 2580
2650 DATA 25,38,37:REM  1981
2660 DATA 15,22,21,21,21:REM  1981
2670 DATA 8,14,12,10,10,10,9,9,9:REM  1981
2680 DATA 5,10,9,8,7,7,6,6,6,6,6,6,6,6,6:REM  1981
2690 DATA 29,47,24:REM  1985
2700 DATA 18,33,25,16,8:REM  1985
2710 DATA 9,19,16,14,12,10,8,6,4,2:REM 1985
2720 DATA 6,12,12,11,10,9,8,7,6,5,4,4,3,2,1:REM  1985
2730 DATA 33,45,22:REM  1986+
2740 DATA 20,32,24,16,8:REM  1986+
2750 DATA 10,18,16,14,12,10,8,6,4,2:REM  1986+
2760 DATA 7,12,12,11,10,9,8,7,6,5,4,3,3,2,1:REM  1986+
2770 WIDTH 80:REM REAL ESTATE DEPRECIATION 1981 LAW
2780 HOME:VTAB 3:HTAB 25:
     PRINT"***** 1981 REAL PROPERTY DEPRECIATION *****":
     VTAB 6: GOTO 2880
2790 HOME:PRINT"YEAR                DEPRECIATION
                        BOOK VALUE"
2800 IF LEFT$(P$,1)="P" OR LEFT$(P$,1)="p" THEN GOTO 2802 ELSE 2810
2802 LPRINT " ":LPRINT T$:LPRINT D$:LPRINT O$:
     LPRINT" ":LPRINT"COST:$";FC
2810 Z$="####         ###,###,###,###.##     ###,###,###,###,###.##"
2820 IF LEFT$(P$,1)="P" OR LEFT$(P$,1)="p" THEN LPRINT "MONTH ";L:
     IF L$="Y" THEN LPRINT "LOW INCOME HOUSING"
2830 IF LEFT$(P$,1)="P" OR LEFT$(P$,1)="p" THEN LPRINT "YEAR
2840 RETURN
2850 X=INT(100*(.01*A(I)*FC)+.5)/100:Y=X+Y:PRINT USING Z$;YR;X;FC-Y
2860 IF LEFT$(P$,1)="P" OR LEFT$(P$,1)="P" THEN
     LPRINT USING Z$;YR;X;FC-Y
2870 RETURN
2880 INPUT "(C)RT, (P)RINTER ";P$
2890 PRINT:INPUT "FIRST COST ";FC
2900 INPUT"YEAR (19XX) ";YR:RESTORE:FOR I=1 TO 99:READ X:NEXT
2910 IF YR<1981 THEN 2900
2920 PRINT"LIFE OF ASSET 15 YEARS ONLY"
```

```
2930 INPUT "MONTH PURCHASED 1,2,... 12";L
2940 IF L<1 OR L>12 THEN 2930
2950 INPUT "LOW-INCOME HOUSING (Y/N)";L$
2960 GOSUB 2790:IF LEFT$(L$,1)="Y" OR LEFT$(L$,1)="Y"  THEN
     FOR I=1 TO 192:READ V:NEXT I
2970 J=0:FOR I=1 TO L*16:J=J+1:READ A(J):IF J=16 THEN J=0
2980 NEXT I:RESTORE:Y=0:FOR I=1 TO 16:GOSUB 2850:YR=YR+1
2990 NEXT I:PRINT:PRINT:HTAB 28:INPUT"<CR> TO CONTINUE";J$:GOTO 90
3000 DATA 12,10,9,8,7,6,6,6,6,5,5,5,5,5,5,0:REM JANUARY
3010 DATA 11,10,9,8,7,6,6,6,6,6,5,5,5,5,5,0:REM FEB
3020 DATA 10,11,9,8,7,6,6,6,6,5,5,5,5,5,5,1:REM MARCH
3030 DATA  9,11,9,8,7,6,6,6,6,6,5,5,5,5,5,1:REM APRIL
3040 DATA 8,11,10,8,7,7,6,6,5,5,5,5,5,5,5,2:REM MAY
3050 DATA 7,11,10,8,7,7,6,6,6,5,5,5,5,5,5,2:REM JUNE
3060 DATA 6,11,10,9,8,7,6,5,5,5,5,5,5,5,5,3:REM JULY
3070 DATA 5,11,10,9,8,7,6,6,5,5,5,5,5,5,5,3:REM AUGUST
3080 DATA 4,11,10,9,8,7,6,6,5,5,5,5,5,5,5,4:REM SEPT
3090 DATA 3,11,10,9,8,7,6,6,6,5,5,5,5,5,5,4:REM OCT
3100 DATA 2,11,10,9,8,7,6,6,6,6,5,5,5,5,5,4:REM NOV
3110 DATA 1,12,10,9,8,7,6,6,6,5,5,5,5,5,5,5:REM DEC
3120 REM FOR LOW INCOME HOUSING
3130 DATA 13,12,10,9,8,7,6,5,5,5,4,4,4,4,4,0:REM JANUARY
3140 DATA 12,12,10,9,8,7,6,5,5,5,5,4,4,4,4,0:REM FEB
3150 DATA 11,12,10,9,8,7,6,5,5,5,5,4,4,4,4,1:REM MARCH
3160 DATA 10,12,10,9,8,7,6,5,5,5,5,5,4,4,4,1:REM APRIL
3170 DATA  9,12,11,9,8,7,6,5,5,5,5,4,4,4,4,2:REM MAY
3180 DATA  8,12,11,9,8,7,6,5,5,5,5,5,4,4,4,2:REM JUNE
3190 DATA 7,12,11,9,8,7,6,5,5,5,5,5,5,4,4,2:REM JULY
3200 DATA 6,13,11,9,8,7,6,5,5,5,5,5,4,4,4,3:REM AUGUST
3210 DATA 4,13,11,10,8,7,6,5,5,5,5,5,5,4,4,3:REM SEPT
3220 DATA 3,13,11,10,8,7,6,5,5,5,5,5,5,5,4,3:REM OCT
3230 DATA 2,13,11,10,8,7,6,6,5,5,5,5,5,4,4,4:REM NOV
3240 DATA 1,13,11,10,9,7,6,6,5,5,5,5,5,4,4,4:REM DEC
3250 REM DISCOUNTED VALUE OF LOAN
3260 HOME:VTAB 3:HTAB 20:
     PRINT"***** DISCOUNTED VALUE OF A BOND/MORTGAGE *****":VTAB 6:
3270 SL$="V":INPUT "(B)OND OR (M)ORTGAGE";SM$
3280 IF LEFT$(SM$,1)="M" OR LEFT$(SM$,1)="M" THEN SM$="C":GOTO 520
3290 INPUT "FACE VALUE OF BOND";VV#
3300 INPUT "STATED RATE OF RETURN";RR#:RR#=.01*RR#
3310 INPUT "LIFE OF BOND";NO%:INPUT "PAYMENTS PER YEAR";YY
3320 INPUT "MARKET RATE OF RETURN";IN#:IN#= .01*IN#:SUM#=0!:
     IN#=IN#/YY:RR#=RR#/YY
3330 NO%=NO%*YY;FOR I=1 TO NO%:SUM#=SUM#+RR#*VV#/(1+IN#)^I
3340 NEXT:SUM#=SUM#+VV#/(1+IN#)^NO%
3350 GOTO 3390
3360 INPUT "MARKET RATE OF RETURN";II#:II#=.01*II#:BP%=MO#
3370 INPUT "BALLOON NOTE (Y/N)";Y$:
     IF LEFT$(Y$,1)="Y" OR LEFT$(Y$,1)="Y" THEN 1410
3380 NO=YR#*PY#:SUM#=PAY# * ((1+II#/PY#)^NO-1 )/
     ((1+II#/PY#)^NO*II#/PY#) + BV#/(1+II#/PY#)^NO
3390 PRINT "CURRENT VALUE: ":PRINT USING A$;SUM#
3400 PRINT:INPUT "<CR> TO CONTINUE";DU$:GOSUB 3880
3410 GOTO 90
3420 REM BREAK-EVEN ANALYSIS
```

```
3430 Z$="#####.#"
3440 HOME:WIDTH 80
3450 PRINT :PRINT :PRINT "BREAK-EVEN ANALYSIS"
3460 PRINT:INPUT "FIXED COST FOR ALT. NO. 1: ";F(1):PRINT:
     INPUT "VARIABLE COST FOR ALT. 1: ";V(1)
3470 PRINT:INPUT "FIXED COST FOR ALT. NO. 2: ";F(2):
     PRINT:INPUT "VARIABLE COST FOR ALT. 2: ";V(2):PRINT
3480 INPUT "PRODUCTION/SALES PER YEAR";RATE
3490 BREAK=(F(2)-F(1))/(V(1)-V(2))
3500 IF RATE>0 THEN PAYBACK=BREAK/RATE ELSE PAYBACK =0
3510 IF BREAK<0 THEN PRINT
     "BREAK-EVEN POINT IS NEGATIVE: ONE   ALTERNATIVE DOMINATES THE
     OTHER-- ";BREAK;:INPUT JUNK$:GOTO 3420
3520 PRINT:INPUT "(C)RT, (P)RINTER ";Y$
3530 HOME
3540 REM SCREEN PRINT WITH PRINT DUMP   *****
3541 IF LEFT$(Y$,1)="P" OR LEFT$(Y$,1)="p" THEN POKE 3,150
3544 FOR COL=0 TO 79:FOR ROW = 0 TO 21:VIDEO%(COL,ROW)=32:
     NEXT ROW:NEXT COL
3545 SR$="B r e a k - E v e n    A n a l y s i s": SSS=8:ROW=0:
     GOSUB 3671
3550 SR$="Break-even Point: ":
3552 SR$="STR$(BREAK):SSS=21:ROW=1:GOSUB 3671
3553 SR$="  Payback: ":SSS=30:ROW=1:GOSUB 3671
3554 SR$="STR$(PAYBACK):SSS=46:ROW=1:GOSUB 3671
3555 SR$="   Y Value: ":SSS=62:ROW=1:GOSUB 3671
3556 SR$=STR$(BREAK*V(1)+F(1)):SSS=73:ROW=1:GOSUB 3671
3560 GOSUB 3640
3570 SR$="STR$(MAX):SSS=2:ROW=4:GOSUB 3671
3580 SR$=STR$(MAX/2):SSS=2:ROW=11:GOSUB 3671
3590 SR$="   0":SSS=2:ROW=18:GOSUB 3671
3595 TA=20:TB=40:IF EIGHT=80 THEN TA=32:TB=64
3600 SR$= "Pieces   0":SSS=0:ROW=19:GOSUB 3671
3601 SR$=STR$(BREAK):SSS=TA:ROW=19:GOSUB 3671
3602 SR$=STR$(BREAK*2):SSS=TB:ROW=19:GOSUB 3671
3610 SR$="Years    0":SSS=0:ROW=20:GOSUB 3671
3611 SR$=STR$(PAYBACK):SSS=TA:ROW=20:GOSUB 3671
3612 SR$=STR$(2*PAYBACK):SSS=TB:ROW=20:GOSUB 3671
3613 GOTO 3680
3620 HOME
3622 FOR ROW=0 TO 21:FOR COL=0 TO EIGHT-1
3623 PRINT CHR$(VIDEO%(COL,ROW));:NEXT COL:PRINT:NEXT ROW
3624 RETURN
3640 MAX=F(1)+V(1)*2*BREAK
3650 MXX=F(2)+V(2)*2*BREAK
3660 IF MXX>MAX THEN MAX=MXX
3670 RETURN
3671 ROW =ROW+1:IF SR$="" THEN RETURN ELSE
     FOR COL=SSS TO SSS+LEN(SR$)-1:CXL=COL-SSS+1:
     VIDEO%(COL,ROW)=ASC(MID$(SR$,CXL,1)):NEXT COL:RETURN
3680 ZZZ=45:SSS=20:IF EIGHT=80 THEN ZZZ=60
3685 INC=2*BREAK/ZZZ:FOR D=1 TO 2:X=10:FOR I=0 TO 2*BREAK STEP INC
3690 VALUE=(V(D)*I+F(D))/MAX
3700 PLTX=SSS-INT(SSS*VALUE)
3710 VIDEO%(X,PLTX)=120-(D-1)*74
```

```
3720 X=X+1
3730 NEXT I
3740 NEXT D
3745 GOSUB 3620
3765 POKE 3,149
3766 PRINT:HTAB 20:
     INPUT "********* PRESS <CR> TO CONTINUE ********";DU$
3770 GOTO 90
3780 LPRINT"ALTERNATIVE","FIXED COST","VARIABLE COST"
3790 GOSUB 3800:GOTO 3760
3800 FOR I=1 TO 2
3810 LPRINT I,F(I),V(I):NEXT I
3820 LPRINT:LPRINT:X=61440!
3830 FOR ROW=1 TO 23:FOR COL=1 TO 80
3840 COX=PEEK(X):X=X+1:LPRINT CHR$(COX);
3850 NEXT COL:LPRINT:X=X+128-80
3860 NEXT ROW
3870 RETURN
3880 IF LEFT$(DU$,1)="P" OR LEFT$(DU$,1)="P" THEN GOSUB 3820
3890 RETURN
4000 REM INITIAL SCREEN PROMO FOR EVALPAC
4010 HOME:VTAB 5:HTAB 20:
     PRINT"***** EVALPAC APPLE CP/M PROGRAM *****":PRINT:
     PRINT:PRINT:PRINT:
4020 HTAB 39:PRINT"by":PRINT:PRINT:PRINT:
4030 HTAB 25:PRINT"Steven M. Zimmerman Ph.D."
4040 HTAB 31:PRINT"Leo Conrad"
4050 HTAB 25:PRINT"Donald R. Smith Ph.D., P.E.":PRINT:PRINT:
     PRINT:PRINT:
4060 HTAB 25:INPUT "Press  < CR > to Begin ";DDDUM$:HOME:
4070 RETURN
4100 REM SUB FOR LOAN ANA SCRN FORMAT
4102 IF T$="P" THEN CCCCC$="Amt of Payment($)" ELSE
     CCCCC$="Reduction"
4120 GOSUB 5000:PRINT TAB(3);"$ Amt. of Loan";TAB(20);
     "A.P.R.(%)";TAB(32);"No. Pay/Yr.";TAB(45);CCCCC$;TAB(68);
     "NO. OF YEARS":
4125 IF T$="P" THEN GOTO 4130 ELSE 4140
4130 PRINT TAB(46); "(0) to Cal."
4140 GOSUB 5000
4150 PRINT TAB(3);:INPUT ;LO#:PRINT TAB(21);:
     INPUT ;APR#: PRINT TAB(32);:INPUT ;PP%:PRINT TAB(43);:
     INPUT ;PAY#:PRINT TAB(66);: INPUT ;YEAR
4155 PRINT:PRINT:
4160 RETURN
5000 REM PLACE DASHES ACROSS SCREEN
5010 FOR ZZZZ%=1 TO 80:PRINT CHR$(45);:NEXT ZZZZ%:RETURN
```

# Chapter 24—CASH.BAS

```
 5 REM "CASH"
10 GOTO 20
11 PRINT:FOR PPPP%=1 TO 80:PRINT CHR$(45);:NEXT PPPP%:PRINT:RETURN
20 WIDTH 80
30 CG=.4:GC=1-CG:REM  1981 CAPITAL GAINS RATE
40 GG$="GO":GX$="GO":GA$="GO":GB$="GO":GC$="GO":DIM TAX(17,3),ZD(15)
50 GOSUB 3000
90 CQ$="## ######## ######### ######### ######## ########## #########
   ######## ######## ########### ###"
92 CQQ$=".## ######## #########   ###########"
94 C$=CQ$+CQQ$
100 CD$="## ######### ######### ######### ####.## ######### ########"
110 CC$="## ######### ######### ####.## ######### #########
    ##########"
120 CX$="## ######### ####.## ######### #########  #########"
130 DIM A$(26):A$(2)="REGULAR  ":A$(3)="CAPITAL   ":A$(4)="OTHER    "
140 A$(5)=A$(3):A$(6)="DEPRECIATION ":A$(7)="MORTGAGE  ":
    A$(8)="INTEREST  "
150 A$(9)="REDUCTION  ":A$(10)="TAXABLE ":A$(11)="TAX        "
160 A$(13)="CASH       ":A$(15)="INCOME    ":A$(16)=A$(15)
170 A$(17)="COSTS      ":A$(18)=A$(17):A$(19)="EXPENSE      ":
    A$(20)=A$(13):A$(21)="EXPENSE    "
180 A$(22)="LOAN       ":A$(23)=A$(16):A$(24)="RATE       "
190 A$(26)="FLOW      ":A$(1)="YEAR ":A$(14)="      ":
    A$(12)="TAX       ":A$(25)="PAID     "
200 PRINT:PRINT:PRINT"CASH FLOW ANALYSIS PROGRAM FOR APPLE CP/M":
    PRINT:PRINT:PRINT"Enter all percents as whole numbers--not as
    decimals":PRINT:PRINT:PRINT:B1$="TITLE :"
210 B2$="DATE :":PRINT B1$;:INPUT T$:PRINT:PRINT B2$;"(XX/XX/XX)";
220 INPUT D$:PRINT:INPUT "STUDY PERIOD IN YEARS";G:
    DIM Z(G,13),X(50):HOME
230 HTAB 34:PRINT"** INCOME MENU **":PRINT:PRINT:
240 GOSUB 11:HTAB 23:PRINT "INPUT MENU REGULAR INCOME":PRINT:
250 IF GG$="GO" THEN FOR I=1 TO G:Z(I,1)=0:NEXT I:GG$="STOP"
260 PRINT "    (C)ONSTANT"
270 PRINT "    (R)EDO ALL REGULAR INCOME"
280 PRINT "    (S)PECIFIED YEAR BY YEAR"
290 PRINT"    (E)ND CYCLE":PRINT:PRINT:HTAB 20:
    INPUT "Enter your choice (C/R/S/E) ";IX$
300 IF LEFT$(IX$,1)="E" OR LEFT$(IX$,1)="E" THEN 420
310 IF LEFT$(IX$,1)="R" OR LEFT$(IX$,1)="R" THEN GG$="GO":GOTO 240
320 IF LEFT$(IX$,1)<>"C" THEN IF  LEFT$(IX$,1)<>"C" THEN 390
330 REM CONSTANT INCOME
340 PRINT:HTAB 4:PRINT "(M)ONTHLY  OR (A)NNUAL VALUES"
342 PRINT:PRINT:HTAB 20:INPUT "Enter Your Choice (M/A) ";L$
350 IF LEFT$(L$,1)="M" OR LEFT$(L$,1)="M" THEN INPUT
    "MONTHLY INCOME";Y:Y=Y*12
360 IF LEFT$(L$,1)="A" OR LEFT$(L$,1)="A"  THEN HTAB 4:PRINT:
    INPUT"YEARLY INCOME";Y
370 FOR I=1 TO G:Z(I,1)=Y+Z(I,1):NEXT
380 GOTO 240
390 PRINT:PRINT:HTAB 25:PRINT"YEAR";TAB(36);"AMOUNT IN  ($)":
    HTAB 25:PRINT"----";TAB(36);"-----------------":PRINT:
392 PRINT TAB(26);:INPUT ;I:PRINT TAB(37);:INPUT ;Y
```

```
400 IF I=0 AND Y=0 THEN 240
410 Z(I,1)=Y+Z(I,1):GOTO 392
420 PRINT:INPUT "VACANCY FACTOR (APPLIED TO ALL VALUES)";VF:
    VF=.01*VF
430 FOR I=1 TO G:Z(I,1)=(1-VF)*Z(I,1):NEXT
440 REM CAP INCOME
442 HOME
450 IF GC$<>"GO" THEN 1630 ELSE PRINT:
    PRINT"        INPUT CAPITAL INCOME YEAR BY YEAR -  ENTER
    0,0 TO TERMINATE"
460 FOR I=1 TO G:Z(I,2)=0:NEXT
470 GOSUB 3200:
500 IF GC$="STOP" THEN 1630
510 PRINT:PRINT:GOSUB 11:HTAB 25:PRINT "INPUT YEARLY COSTS MENU":
    PRINT:PRINT:
520 IF GX$="GO" THEN FOR I=1 TO G:Z(I,3)=0:NEXT:GX$="STOP"
530 PRINT"   (C)ONSTANT"
540 PRINT"   (S)PECIFIED YEAR BY YEAR"
550 PRINT"   (R)EDO ALL YEARLY COSTS"
560 PRINT"   (E)ND CYCLE":PRINT:PRINT:HTAB 25:
    INPUT "Enter your choice (C/S/R/E) ";IX$
570 IF LEFT$(IX$,1)="E" OR LEFT$(IX$,1)="E" THEN 670
580 IF LEFT$(IX$,1)="R" OR LEFT$(IX$,1)="R" THEN GX$="GO":GOTO 510
590 IF LEFT$(IX$,1)="S" OR LEFT$(IX$,1)="S" THEN 630
600 INPUT"INPUT YEARLY COST";C
610 FOR I=1 TO G:Z(I,3)=Z(I,3)+C:NEXT
620 GOTO 510
630 REM YR SPEC
640 HTAB 25:PRINT"YEAR";TAB(37);"AMOUNT IN ($)":HTAB 25:
    PRINT"----";TAB(37);"-------------":PRINT:
642 PRINT TAB(26);:INPUT ;I: PRINT TAB(38);:INPUT Y:
650 IF I=0 AND Y=0 THEN 510
660 Z(I,3)=Z(I,3)+Y:GOTO 642
670 IF GC$<>"GO" THEN 1630
680 GOSUB 11:HTAB 25:PRINT"DEPRECIATION MENU":PRINT:PRINT:
690 IF GA$="GO" THEN FOR I=0 TO G:Z(I,4)=0:Z(I,5)=0:NEXT:
    FOR I=1 TO 50:X(I)=0:NEXT:GA$="STOP":D5=0
700 FOR I=1 TO 50:X(I)=0:NEXT I
710 PRINT"   (L)INE -  STRAIGHT LINE DEPRECIATION"
720 PRINT"   (D)ECLINING BALANCE DEPRECIATION"
730 PRINT"   (S)UM OF THE YEARS DIGITS"
740 PRINT"   (R)EDO ALL DEPRECIATION CALCULATIONS"
750 PRINT"   (T)ABLE LOOKUP 1981"
760 PRINT"   (E)ND CYCLE":
762 PRINT:PRINT:HTAB 25:INPUT "Enter your Choice ";IX$
770 IF LEFT$(IX$,1)="E" OR LEFT$(IX$,1)="E" THEN 1220
780 IF LEFT$(IX$,1)="R" OR LEFT$(IX$,1)="R" THEN GA$="GO":GOTO 680
790 W$(1)="YEAR OF EXPENSE (COUNT 0 AS START YEAR)"
800 W$(2)="CAPITAL EXPENSE"
810 W$(3)="SALVAGE VALUE (LAND VALUE)"
820 W$(4)="LIFE"
830 D(5)=1
840 IF LEFT$(IX$,1)="D" OR LEFT$(IX$,1)="D" THEN D(5)=2
850 IF LEFT$(IX$,1)="T" OR LEFT$(IX$,1)="T" THEN D(5)=5
860 IF LEFT$(IX$,1)="S" OR LEFT$(IX$,1)="S"THEN D(5)=3
```

```
870 W$(7)="RATE"
880 W$(8)="DEPRECIATION"
890 D(8)=100
900 FOR I=1 TO 50:X(I)=0:NEXT I
910 D(4)=15:PRINT "INPUT FOR NEXT ASSET ":FOR I=1 TO 3:
    PRINT W$(I);:INPUT D(I):NEXT I:
915 IF LEFT$(IX$,1)<>"T" THEN IF LEFT$(IX$,1)<> "T" THEN
    PRINT W$(4);:INPUT D(4)
920 Z(G,5)=Z(G,5)+D(3):Z(D(1),4)=Z(D(1),4)+D(2):D(1)=D(1)+1
930 IF D(5)=2 THEN PRINT W$(7);:INPUT D(7):D(7)=D(7)*.01
940 IF D(5)=1 THEN I=1:GOTO 1070
950 IF LEFT$(IX$,1)="T" OR LEFT$(IX$,1)="T" THEN 1110
960 IF D(5)=2 THEN 1010
970 REM SYD
980 SD=(D(4)*(D(4)+1))/2:BV=D(2)
990 FOR I=1 TO D(4):DX=((D(4)-(I-1))/SD)*(BV-D(3)):SU=SD+DX:
    D(2)=D(2)-DX:K%=I:X(I)=DX
1000 D(1)=D(1)+1:NEXT I:GOTO 1180
1010 REM DECL
1020 DR=D(7)/D(4)
1030 FOR I=1 TO D(4):DX=DR*D(2):SX=(D(2)-D(3))/(D(4)-I+1):
     IF D(3)>(D(2)-DX) THEN DX=D(2)-D(3):GOTO 1050
1040 IF SX>DX THEN 1070
1050 X(D(1))=X(D(1))+DX:SU=SU+DX:D(2)=D(2)-DX
1060 D(1)=D(1)+1:NEXT I
1070 REM STR
1080 IF (D(4)-I+1)>0 THEN DX=(D(2)-D(3))/(D(4)-I+1) ELSE 1090
1090 FOR J=I TO D(4):SU=SU+DX:D(2)=D(2)-DX:X(D(1))=DX+X(D(1))
1100 D(1)=D(1)+1:NEXT J:GOTO 1180
1110 REM 1981 LAW
1120 INPUT "LOW INCOME HOUSING (Y/N)";D7$:
     IF LEFT$(D7$,1)="Y" OR LEFT$(D7$,1)="Y" THEN FOR I=1 TO 15:
     READ JU:NEXT I
1130 FOR I=1 TO 15:READ ZD(I):NEXT I
1140 DATA 12,10,9,8,7,6,6,6,6,5,5,5,5,5,5:REM REG
1150 DATA 13,12,10,9,8,7,6,5,5,5,4,4,4,4,4:REM LOW
1160 RESTORE
1170 FOR I=1 TO 15:X(I)=ZD(I)*(D(2)-D(3))/100:NEXT I
1180 D5=D5+X(G):FOR I=1 TO G:Z(I,5)=Z(I,5)+X(I):NEXT I
1190 KK=G:IF KK>D(4) THEN KK=D(4)
1200 IF D(4)>G THEN S=0:FOR I=G+1 TO D(4):S=S+X(I):NEXT I:
     Z(G,5)=Z(G,5)+S
1210 GOTO 680
1220 IF GB$<>"GO" THEN 1630
1230 GOSUB 11:PRINT"MORTGAGE MENU":PRINT:PRINT:
1240 IF GB$="GO" THEN GB$="STOP":FOR I=0 TO G:Z(I,8)=0:Z(I,6)=0:
     Z(I,7)=0:NEXT I
1250 PRINT"    (P)AYMENTS EQUAL"
1260 PRINT"    (D)EDUCTION EQUAL"
1270 PRINT"    (I)NTEREST ONLY"
1280 PRINT"    (R)EDO ALL MORTGAGES"
1290 INPUT"    (E)ND CYCLE";IX$
1300 IF LEFT$(IX$,1)="E" OR LEFT$(IX$,1)="E" THEN 1630
1310  X2$="START YR, LOAN, APR, PAYMENTS/YR, PAYMENT, YRS"
1320 IF LEFT$(IX$,1)="R" OR LEFT$(IX$,1)="R" THEN GB$="GO":GOTO 1230
```

```
1330 IF LEFT$(IX$,1)="P" OR LEFT$(IX$,1)="P" THEN INPUT
     "BALLOON YEAR (1000 IF NONE)";BA
1340 IF LEFT$(IX$,1)<>"P" THEN IF LEFT$(IX$,1)<>"P" THEN 1480
1350 PRINT"INPUT 0 TO CALCULATE PAYMENT"
1360 PRINT:PRINT TAB(4);"YEAR";TAB(13);"LOAN AMOUNT ($)";
     TAB(32);"APR (%)";TAB(42);"PAYMENTS/YR.";TAB(56);"PAYMENTS";
     TAB(69);"YEARS":FOR QQ%=1 TO 80:PRINT CHR$(45);:NEXT QQ%:PRINT:
1362 PRINT TAB(5);:INPUT ;L:PRINT TAB(15);:INPUT ;LO:
     PRINT TAB(33);:INPUT ;APR:PRINT TAB(45);:INPUT ;PP:
     PRINT TAB(57);:INPUT ;PAY:PRINT TAB(70);:INPUT ;YEAR:PRINT:
1364 Z(L,6)=Z(L,6)+LO:APR=.01*APR
1370 R=APR/PP:IF PAY=0 THEN II=(1+R)^(PP*YEAR):FF=(R*II)/(II-1):
     PAY=FF*LO
1380 N=YEAR*PP:KK=G:IF YEAR<G THEN KK=YEAR
1390 J=0:JL=0
1400 J=J+1:S=0:SS=0:IF BA+1=J THEN YY=BA+L:GOTO 1460
1410 IF JL>=G THEN YY=KK:GOTO 1460
1420 FOR I=1 TO PP
1430 II=R*LO:RED=PAY-II:LO=LO-RED
1440 S=S+II:SS=SS+RED
1450 NEXT I:YY=KK:JL=J+L:Z(JL,7)=Z(JL,7)+S:Z(JL,8)=Z(JL,8)+SS:
     IF J<KK THEN 1400
1460 IF LO>0 THEN Z(YY,8)=Z(YY,8)+LO
1470 GOTO 1230
1480 IF LEFT$(IX$,1)<>"D" THEN IF LEFT$(IX$,1)<>"D" THEN 1560
1490 PRINT"START YEAR, LOAN, APR, REDUCTION PER YEAR, YEARS":
     INPUT L,LO,APR,RED,YEAR:Z(L,6)=Z(L,6)+LO:APR=.01*APR
1500 PP=1:KK=G:IF YEAR<KK THEN KK=YEAR
1510 FOR I=L+1 TO KK:II=LO*APR:LO=LO-RED:Z(I,7)=Z(I,7)+II:
     Z(I,8)=Z(I,8)+RED:IF LO<0 THEN RED=RED-LO:Z(I,8)=Z(I,8)-LO:
     GOTO 1530
1520 NEXT I
1530 IF KK=YEAR THEN KK=KK-1
1540 IF LO>0 THEN Z(KK,8)=Z(KK,8)+LO
1550 GOTO 1230
1560 REM I/O
1570 IF LEFT$(IX$,1)<>"I" THEN IF LEFT$(IX$,1)<>"I" THEN 1230
1580 PRINT"START YEAR, LOAN, APR, YEARS":PP=1:INPUT L,LO,APR,YEAR:
     Z(L,6)=Z(L,6)+LO:APR=APR*.01
1590 KK=G:IF KK>YEAR THEN KK=YEAR
1600 FOR I=L+1 TO KK:Z(I,7)=Z(I,7)+APR*LO:NEXT I
1610 Z(KK,8)=Z(KK,8)+LO
1620 GOTO 1230
1630 GC$="STOP"
1640 HOME:VTAB 3:HTAB 15:PRINT"***** TAX CATEGORIES *****":PRINT:PRINT:
1650 PRINT "      (C)ORPORATION"
1660 PRINT "      (I)NDIVIDUAL"
1670 INPUT SS$:IF LEFT$(SS$,1)="C" OR LEFT$(SS$,1)="C" THEN
     SS=2 ELSE SS=1
1680 PRINT A$(11),A$(24);:INPUT TR:TR=TR*.01:FOR I=1 TO G:
     Z(I,10)=TR:NEXT I
1690 IF SS<1 OR SS>3 THEN 1640
1700 GOSUB 11:HTAB 30:PRINT"*** OUTPUT MENU ***":PRINT:PRINT:
1710 PRINT"     (C)RT"
1720 PRINT"     (40) COLUMN PRINTER"
```

```
1730 PRINT"    (80) COLUMN PRINTER"
1740 PRINT"    (130) COLUMN PRINTER":
1742 PRINT:PRINT:HTAB 20:INPUT "Enter Your Choice (C/40/80/130) ";IX$
1750 IF LEFT$(IX$,1)="C" OR LEFT$(IX$,1)="C" THEN OP=1
1760 IF LEFT$(IX$,1)="8" THEN OP=2:GOSUB 3100
1770 IF LEFT$(IX$,1)="1" THEN OP=3:GOSUB 3100
1780 IF LEFT$(IX$,1)="4" THEN OP=4:GOSUB 3100
1790 PRINT:PRINT"NUMBER OF YEARS OF STUDY <=";G;:INPUT YS:
     H$="INTERNAL RATE":PRINT:PRINT H$;:INPUT R:H1$="INFLATION":R=.01*R
1800 IF YS>G THEN 1790
1810 GOSUB 11:HTAB 25:PRINT"INFLATION MENU":PRINT:PRINT:
1820 PRINT"   (S)INGLE RATE ALL FACTORS"
1830 PRINT"   (I)NDIVIDUAL RATES"
1840 INPUT"       REG. COSTS/CAPITAL COSTS";IX$
1850 IF LEFT$(IX$,1)="I" OR LEFT$(IX$,1)="I" THEN 1880
1860 PRINT H1$;:INPUT M:M=.01*M
1870 FOR I=1 TO 4:IM(I)=M:NEXT I: GOTO 1930
1880 REM INFLATE
1890 INPUT"INPUT INFLATION RATE FOR REGULAR INCOME";IM(1):M=-9999
1900 INPUT"INPUT INFLATION RATE FOR CAPITAL INCOME";IM(2)
1910 INPUT"INPUT INFLATION RATE FOR REGULAR COSTS";IM(3)
1920 INPUT"INPUT INFLATION RATE FOR CAPITAL COSTS";IM(4):
     FOR I=1 TO 4:IM(I)=IM(I)*.01:NEXT I
1930 HOME:IF OP>1 THEN LPRINT" ":IF M>=0 THEN LPRINT B1$;
     T$;" ";H1$;M ELSE LPRINT B1$;T$;" ";H1$;IM(1);IM(2);IM(3);IM(4)
1940 IF OP>1 THEN LPRINT B2$;D$,H$;R:LPRINT C$(SS):LPRINT" "
1950 IF OP>1 AND SS=3 THEN LPRINT VV$(ST)
1960 IF OP=4 THEN FOR I=1 TO 5:LPRINT A$(I);:NEXT:LPRINT:
     FOR I=14 TO 18:LPRINT A$(I);:NEXT:LPRINT
1970 IF OP=3 THEN FOR I=1 TO 13:LPRINT A$(I);:NEXT I:
     LPRINT "ADJUSTED":FOR I=14 TO 26:LPRINT A$(I);:NEXT I:
     LPRINT "$ FLOW"
1980 IF OP=2 THEN FOR I=1 TO 8:LPRINT A$(I);:NEXT I:LPRINT:
     FOR I=14 TO 21:LPRINT A$(I);:NEXT I:LPRINT
1990 IF OP=1 THEN FOR I=1 TO 7:PRINT A$(I);:NEXT I:PRINT A$(8):
     FOR I=14 TO 20:PRINT A$(I);:NEXT I:PRINT A$(21)
2000 RR=Z(0,6)-Z(0,4):IF OP=3 THEN
     LPRINT USING C$;0,0,0,0,Z(0,4),0,Z(0,6),0,0,0,0,0,RR,RR
2010 IF OP=1 THEN PRINT USING C$;0,0,0,0,Z(0,4),0,Z(0,6),0
2020 IF OP=2 THEN LPRINT USING C$;0,0,0,0,Z(0,4),0,Z(0,6),0
2030 IF OP=4 THEN LPRINT USING C$;0,0,0,0,Z(0,4)
2040 ST=0
2050 REM INFLAT
2060 SR=RR:ST=0:FOR I=1 TO YS-1:FOR J=1 TO 4:IN(J)=(1+IM(J))^I:
     NEXT J:Z(I,9)=IN(1)*Z(I,1)-IN(3)*Z(I,3)-Z(I,5)-Z(I,7):
     IF I=1 THEN Z(1,9)=Z(1,9)+Z(0,9)
2070 IF SS=1 THEN Z(I,11)=CG*Z(I,10)*(IN(2)*Z(I,2)-IN(4)*Z(I,4))
     ELSE Z(I,11)=0
2080 D6=Z(I,10)*Z(I,9):K%=I/13:K%=K%*13:KK%=I:IF SS <>1 THEN
     Z(I,9)=Z(I,9)+IN(2)*Z(I,2)
2090 IF S=1 AND  Z(I,11)<0 THEN PRINT
     "CAPITAL LOSS - TAXABLE INCOME ADJUSTED":Z(I,9)=Z(I,9)+
     Z(I,11)/Z(I,10)
2100 IF D6>0 THEN Z(I,11)=Z(I,11)+D6
```

```
2110 Z(I,12)=Z(I,9)-Z(I,11)-Z(I,4)*IN(4)+Z(I,5)-Z(I,8)+Z(I,6):
     IF SS=1 THEN Z(I,12)=Z(I,12)+Z(I,2)*IN(2)
2120 X=(1+R)^I:SR=SR+Z(I,12)/X:IF Z(I,9)<0 THEN
     ST=ST+Z(I,9)*Z(I,10)/X
2130 IF OP=1 THEN PRINT USING C$;I,Z(I,1)*IN(1),Z(I,2)*IN(2),
     Z(I,3)*IN(3),Z(I,4)**(4),Z(I,5),Z(I,6),Z(I,7):
     IF K%=KK% THEN INPUT "<CR> TO CONTINUE";R$
2140 IF OP=2 THEN LPRINT USING C$;I,Z(I,1)*IN(1),Z(I,2)*IN(2),
     Z(I,3)*IN(3),Z(I,4))N(4),Z(I,5),Z(I,6),Z(I,7)
2150 Z(I,13)=Z(I,12):IF Z(I,9)<0 THEN Z(I,13)=Z(I,13)-Z(I,10)*Z(I,9)
2160 IF OP=3 THEN LPRINT USING C$;I,Z(I,1)*IN(1),Z(I,2)*IN(2),
     Z(I,3)*IN(3),Z(I,4))N(4),Z(I,5),Z(I,6),Z(I,7),Z(I,88Z(I,9),
     Z(I,10),Z(I,11),Z(I,12),,I,13)
2170 IF OP=4 THEN LPRINT USING C$;I,Z(I,1)*IN(1),Z(I,2)*IN(2),
     Z(I,3)*IN(3),Z(I,4))N(4)
2180 NEXT I
2190 FOR I=1 TO 3:E(I)=0:NEXT I
2200 FOR I=YS TO G:E(1)=E(1)+Z(I,2):E(2)=E(2)+Z(I,5):
     E(3)=E(3)+Z(I,8):NEXT I
2210 FOR J=1 TO 4:IN(J)=(1+IM(J))^(I-1):NEXT J:
     I=YS:Z(I,9)=IN(1)*Z(I,1)-IN(3)*Z(II3)-D5-Z(I,7)
2220 IF SS=1 THEN Z(I,11)=CG*Z(I,10)*(IN(2)*E(1)-E(2)+D5-IN(4)*Z(I,4))
     ELSE Z(I,11)=0
2230 D6=Z(I,10)*Z(I,9):IF SS <>1 THEN
     Z(I,9)=Z(I,9)+IN(2)*Z(I,2)-Z(I,5):Z(I,11)=ZZI,9)*Z(I,10):
     IF Z(I,11)<0 THEN Z(I,11)=0
2240 IF SS=1 AND Z(I,11)<0 THEN PRINT
     "CAPITAL LOSS - TAXABLE INCOME ADJUSTED":
     Z(I,9)=Z(I,9)+Z(I,11)/Z(I,10)
2250 IF D6>0 THEN Z(I,11)=Z(I,11)+D6
2260 Z(I,12)=Z(I,9)+E(2)-E(3)-Z(I,11)+Z(I,6):IF SS=1 THEN
     Z(I,12)=Z(I,9)+D5-E(3)-Z(I,11)+E(1)*IN(2)+Z(I,6)
2270 X=(1+R)^I:SR=SR+Z(I,12)/X:IF Z(I,9)<0 THEN ST=ST+Z(I,9)*Z(I,10)/X
2280 IF OP=1 THEN PRINT USING C$;
     I,Z(I,1)*IN(1),E(1)*IN(2),Z(I,3)*IN(3),Z(I,4)*INN),E(2),
     Z(I,6),Z(I,7)
2290 IF OP=2 THEN LPRINT USING C$;I,Z(I,1)*IN(1),E(1)*IN(2),
     Z(I,3)*IN(3),Z(I,4)*II4),E(2),Z(I,6),Z(I,7)
2300 Z(I,13)=Z(I,12):IF Z(I,9)<0 THEN Z(I,13)=Z(I,13)-Z(I,9)*Z(I,10)
2310 IF OP=3 THEN LPRINT USING C$;I,Z(I,1)*IN(1),E(1)*IN(2),
     Z(I,3)*IN(3),Z(I,4)*II4),E(2),Z(I,6),Z(I,7),E(3),Z(1,9),
     Z(1,10),Z(1,11),Z(1,12),Z(1,13)
2320 IF OP=4 THEN LPRINT USING C$;I,Z(I,1)*IN(1),E(1)*IN(2),
     Z(I,3)*IN(3),Z(I,4)*IN(4)
2330 IF G>7 AND  OP=1 THEN INPUT"<CR> TO CONTINUE";DU$
2340 IF OP=2 THEN LPRINT" "
2350 IF OP=1 THEN PRINT A$(1);:FOR I=9 TO 13:PRINT A$(I);:NEXT I:
     PRINT "  ADJUSTED ":PRINT A$(14);:FOR I=22 TO 26:PRINT A$(I);:
     NEXT I:PRINT"   $ FLOW"
2360 IF OP=2 THEN LPRINT A$(1);:FOR I=9 TO 13:LPRINT A$(I);:NEXT I:
     LPRINT"  ADJUSTED ":LPRINT A$(14);:FOR I=22 TO 26:
     LPRINT A$(I);:NEXT I:LPRINT "   $ FLOW"
2370 IF OP=4 THEN LPRINT " ":LPRINT A$(1);:FOR I=6 TO 9:
     LPRINT A$(I);:NEXT I:LPRINT:LPRINT A$(14);:FOR I= 19 TO 22:
     LPRINT A$(I);:NEXT I:LPRINT
```

```
2380 IF OP=1 THEN PRINT USING CC$;0,0,0,0,0,RR,RR
2390 IF OP=2 THEN LPRINT USING CC$;0,0,0,0,0,RR,RR
2400 IF OP=4 THEN LPRINT USING C$;0,0,Z(0,6),0,0
2410 K=0:FOR I=1 TO YS:K=K+1:EE=Z(I,8):IF I=YS THEN EE=E(3)
2420 IF OP=1 THEN PRINT USING CC$;I,EE,Z(I,9),Z(I,10),Z(I,11),
     Z(I,12),Z(I,13)
2430 IF OP=2 THEN LPRINT USING CC$;I,EE,Z(I,9),Z(I,10),Z(I,11),
     Z(I,12),Z(I,13)
2440 IF OP=4 THEN LPRINT USING C$;I,Z(I,5),Z(I,6),Z(I,7),Z(I,8)
2450 IF OP=1 THEN IF K=23 THEN K=0:INPUT"<CR> TO CONTINUE";DU$
2460 NEXT I
2470 IF OP<>4 THEN 2520
2480 LPRINT " ":LPRINT A$(1);:FOR I=10 TO 13:LPRINT A$(I);:
     NEXT I:LPRINT:LPRINT A$(14);:FOR I=23 TO 26:LPRINT A$(I);:
     NEXT I:LPRINT
2490 LPRINT USING CX$;0,0,0,0,RR
2500 FOR I=1 TO YS:K=K+1:EE=Z(I,8):IF I=YS THEN EE=E(3)
2510  LPRINT USING CX$;I,Z(I,9),Z(I,10),Z(I,11),Z(I,12):NEXT
2520 H$="DISCOUNT CASH":H1$="DISCOUNT TAX SAVINGS":
     H2$="DISCOUNT CASH AND TAX SAVINGS ":T=SR-ST:
     IF OP=1 THEN PRINT :PRINT H$;SR:PRINT H1$;ST:PRINT H2$;T
2530 IF OP>1 THEN LPRINT H$;SR:LPRINT H1$;ST:LPRINT H2$,T
2540 IF OP=1 THEN INPUT"<CR> TO CONTINUE";R$
2550 SR=0:ST=0:IF OP>1 THEN LPRINT:
     LPRINT "*** NOTE: THE AMT. IN THE LAST ROW OF DEPR EXP
     INCLUDES THE RETURN OF CAPITAL":LPRINT
2552 FOR COM%= 1 TO 4:LINE INPUT "COMMENTS FOR PRINTOUT...";A$:
     LPRINT A$:NEXT COM%:LPRINT CHR$(12)
2560 GOSUB 11:HTAB 25:PRINT"RECYCLE MENU":PRINT:PRINT:
2570 PRINT "    (R)EGULAR INCOME"
2580 PRINT"    (C)APITAL INCOME"
2590 PRINT"    (Y)EARLY EXPENSES"
2600 PRINT"    (D)EPRECIATION EXPENSE"
2610 PRINT"    (M)ORTGAGES AND LOANS"
2620 PRINT"    (I)NFLATION AND OUTPUT OPTIONS"
2630 PRINT"    (E)ND PROGRAM"
2640 PRINT "    ";:INPUT IX$
2650 IF LEFT$(IX$,1)="E" OR LEFT$(IX$,1)="E" THEN END
2660 IF LEFT$(IX$,1)="R" OR LEFT$(IX$,1)="R" THEN 230
2670 IF LEFT$(IX$,1)="C" OR LEFT$(IX$,1)="C" THEN 460
2680 IF LEFT$(IX$,1)="Y" OR LEFT$(IX$,1)="Y" THEN 510
2690 IF LEFT$(IX$,1)="D" OR LEFT$(IX$,1)="D" THEN 680
2700 IF LEFT$(IX$,1)="M" OR LEFT$(IX$,1)="M" THEN 1230
2710 IF LEFT$(IX$,1)="I" OR LEFT$(IX$,1)="I" THEN 1640
2720 GOTO 2560
2730 REM TAX SUB
2740 INC=OI+Z(I,9)
2750 IF INC<0 THEN Z(I,10)=0:RETURN
2760 IF RB$="R" THEN 2800
2770 FOR K=1 TO ZZ-1:Z(I,10)=TAX(K-1,3)
2780 IF INC>TAX(K-1,1) THEN IF INC<TAX(K,1) THEN RETURN
2790 NEXT K:RETURN
2800 SX=0:SUM=0:FOR K=1 TO ZZ-1:Z(I,10)=TAX(K-1,3):
     SUM=SUM+TAX(K,2):SX=SX+TAX(K,1)
2810 IF INC>TAX(K-1,1) THEN IF INC<TAX(K,1) THEN 2830
```

```
2820 NEXT K:K=ZZ
2830 DT=INC-SX:IF DT<0 THEN 2870
2840 SUM=SUM+DT*Z(I,10)
2850 Z(I,10)=SUM/INC:IF Z(I,10)<0 THEN Z(I,10)=0
2860 RETURN
2870 Z(I,10)=SUM/SX:IF Z(I,10)<0 THEN Z(I,10)=0
2880 RETURN
3000 REM INITIAL PROMO SCREEN SUBROUTINE
3010 HOME:VTAB 5:HTAB 17:
     PRINT"**** APPLE CP/M CASH FLOW ANALYSIS PROGRAM ****":
     VTAB 8:HTAB 39:PRINT"by":VTAB 12:
3020 HTAB 25:PRINT"Steven M. Zimmerman Ph.D.":
3030 HTAB 31:PRINT"Leo Conrad"
3040 HTAB 25:PRINT"Donald R. Smith Ph.D., P.E.":
3050 VTAB 20:HTAB 25:INPUT "Press  <CR> to Begin";DU$:HOME:RETURN
3100 REM SUB TO GET PRINTER READY
3110 PRINT:PRINT:PRINT CHR$(7):
     PRINT"*** ALIGN PAPER IN PRINTER & TURN ON ...<CR> WHEN READY"
3120 HTAB 35:INPUT DUMMM$:PRINT:PRINT:RETURN
3200 REM CAPITAL INCOME DATA ENTRY
3210 PRINT:PRINT:PRINT::GOSUB 11:HTAB 25:
     PRINT"*** CAPITAL INCOME DATA ENTRY ***":
3220 PRINT:HTAB 27:PRINT"( Enter 0,0 to Terminate )":PRINT:PRINT:
3230 HTAB 25:PRINT"Year";TAB(32);"Capital Income by Year":
3240 HTAB 25:PRINT"----";TAB(32);"----------------------":PRINT:
3250 PRINT TAB(26);:INPUT ;I:PRINT TAB(34);:INPUT ;Y:Z(I,2)=Y
3260 IF I=0 AND Y=0 THEN RETURN
3270 GOTO 3250
```

# Chapter 25—FORECAST

```
10 REM "FORECAST.BAS"
15 CLEAR 500:EIGHT=80
20 WIDTH 80:GOSUB 2000:
30 HOME:GOTO 140
40 PRINT"*** WARNING: USE CODED DATA WHENEVER POSSIBLE."
60 PRINT
70 HTAB 15:INPUT"DETAILS (Y/N)";Y$:IF LEFT$(Y$,1)="N" OR
   LEFT$(Y$,1)="N" THEN RETURN
80 HOME
90 HTAB 10:PRINT "REGRESSION ANALYSIS REQUIRES THE USE OF SOME
   VERY LARGE NUMBERS"
100 HTAB 10:PRINT "AND THEN FINDING THE DIFFERENCE BETWEEN
    THESE LARGE NUMBERS. IT"
110 HTAB 10:PRINT "IS GOOD PRACTICE TO USE CODED DATA WHENEVER
    POSSIBLE."
120 PRINT
130 PRINT:HTAB 30:INPUT "<CR> TO CONTINUE";D$:RETURN
140 HTAB 15:PRINT "REGRESSION WITH PREDICTION & CONFIDENCE LIMITS"
150 PRINT
160 HTAB 25:PRINT"LEAST SQUARES APPROACH"
170 PRINT
180 HTAB 28:PRINT"f(X)=  a + b * X "
190 PRINT:GOSUB 40
210 I=0
220 PRINT
230 INPUT "NUMBER OF DATA POINTS";G
240 DIM X(G),F(G)
245 DIM VIDEO%(80,22)
260 HOME: PRINT:PRINT:
270 HTAB 25:PRINT"*** DATA INPUT MENU ***":PRINT:PRINT:HTAB 10:
    PRINT"(D)isk":HTAB 10:PRINT"(K)eyboard":HTAB 10:PRINT"(R)ead":
    PRINT:HTAB 5:INPUT "Enter your choice ( D/K/R ) ";IN$
280 IF LEFT$(IN$,1)="D" OR LEFT$(IN$,1)="d" THEN PRINT:PRINT:
    INPUT "DATA FILE NAME = ";F$:PRINT:INPUT "DISK DIVE (A/B/C/D) =";
    D$:D$=D$+":"+F$:OPEN "I",1,D$
290 I=I+1
300 IF LEFT$(IN$,1)="R" OR LEFT$(IN$,1)="R" THEN READ X(I),F(I)
310 IF LEFT$(IN$,1)="D" OR LEFT$(IN$,1)="D" THEN INPUT #1,X(I),F(I)
320 IF LEFT$(IN$,1)="K" OR LEFT$(IN$,1)="k" THEN HOME:GOSUB 2300
330 IF F(I)<>9999 AND X(I)<>9999 THEN 290
340 HOME
350 N=I-1
360 CLOSE:GOTO 440
370 JJJ=42:SSSS=2:WIDTH 80:IF EIGHT=50 THEN WIDTH 128:
    PRINT "I","X","F(X)","I","X","F(X)":GOTO 380
375 PRINT "I","X","F(X)":SSSS=1:JJJ=20
380 K=0
385 IF T>N THEN T=N
```

```
390 FOR I=F TO T STEP SSSS
400 PRINT I,X(I),F(I),;:IF EIGHT=50 THEN PRINT I+1,X(I+1),
    F(I+1) ELSE PRINT
410 K=K+2
420 IF K=JJJ THEN INPUT"<CR> TO CONTINUE";DU$:K=0
430 NEXT
440 PRINT:PRINT:HTAB 34:PRINT "OUTPUT MENU":PRINT:PRINT:
450 PRINT"(A)dd values  (C)hange  (D)elete  (E)nd  (F)ile-disk
    (I)nsert (L)ist  (M)onitor"
452 HTAB 22:PRINT"  (P)rinter  (X)merge data files"
460 PRINT:HTAB 28:INPUT "Enter your choice- ";NO$:I=0
465 WIDTH 80
470 IF LEFT$(NO$,1)="E" OR LEFT$(NO$,1)="E" THEN END
480 IF LEFT$(NO$,1)="L" OR LEFT$(NO$,1)="L" THEN INPUT
    "FROM ";F:INPUT "TO ";T:GOTO 370
490 IF LEFT$(NO$,1)="C" OR LEFT$(NO$,1)="C" THEN INPUT
    "NUMBER TO CHANGE";NO:INPUT"X AND F(X)";X(NO),F(NO):GOTO 440
500 IF LEFT$(NO$,1)="A" OR LEFT$(NO$,1)="A" THEN I=N:IN$="K":GOTO 290
510 IF LEFT$(NO$,1)="M" OR LEFT$(NO$,1)="M" THEN 710
520 IF LEFT$(NO$,1)="P" OR LEFT$(NO$,1)="P" THEN 710
530 IF LEFT$(NO$,1)="D" OR LEFT$(NO$,1)="D" THEN 620
540 IF LEFT$(NO$,1)="I" OR LEFT$(NO$,1)="I" THEN 660
550 IF LEFT$(NO$,1)="X" OR LEFT$(NO$,1)="X" THEN 700
560 I=0
570 IF LEFT$(NO$,1)="F" OR LEFT$(NO$,1)="F" THEN
    INPUT "DATA FILE NAME :";F$:INPUT  "DISK (A,B,C,D)";D$:
    D$=D$+":"+F$:OPEN "O",2,D$ ELSE GOTO 450
580 I=I+1:PRINT #2, X(I),F(I)
590 IF F(I)<>9999 AND X(I)<>9999 THEN 580
600 CLOSE
610 GOTO 440
620 INPUT "NUMBER TO BE DELETED (0 TO EXIT)";L
630 IF L=0 THEN 450
640 FOR I=L TO N:X(I)=X(I+1):F(I)=F(I+1):NEXT I
650 X(N)=9999:F(N)=9999:N=N-1:GOTO 440
660 INPUT "NUMBER TO BE INSERTED (0 TO EXIT)";L
670 IF L=0 THEN 450 ELSE INPUT "X AND F(X)"; A,B
680 FOR I=N TO L STEP -1:X(I+1)=X(I):F(I+1)=F(I):NEXT I
690 X(L)=A:F(L)=B:N=N+1:GOTO 440
700 I=N:IN$="D":GOTO 280
710 SX#=0: SY#=0:S2#=0:SC#=0
720 TX=0:TY=0:BX=99999!
730 BY=99999!:S3#=0: S4#=0:S5#=0:SA#=0
740 FOR I=1 TO N
750 SX#=SX#+X(I): SY#=SY#+F(I)
760 SC#=SC#+F(I)*X(I)
770 S2#=S2#+X(I)*X(I): S3#=S3#+X(I)*X(I)*X(I)
780 S4#=S4#+X(I)*X(I)*X(I)*X(I): S5#=S5#+F(I)*X(I)*X(I):
    SA#=SA#+F(I)*F(I)
790 IF X(I)<BX THEN BX=X(I)
800 IF X(I)>TX THEN TX=X(I)
810 IF F(I)<BY THEN BY=F(I)
820 IF F(I)>TY THEN TY=F(I)
830 NEXT:ZX=N*S2#-SX#*SX#
840 ZY=N*SA#-SY#*SY#
```

```
850 ZZ=N*SC#-SX#*SY#:SE=SQR((ZX*ZY-ZZ*ZZ)/(N*(N-2)*ZX))
860 IF LEFT$(NO$,1)="P" OR LEFT$(NO$,1)="P" THEN 870 ELSE 940
870 LPRINT " REGRESSION  WITH PREDICTION LIMITS LEAST SQUARES APPROACH"
880 LPRINT "  F(X)= A + B * X   "
890 INPUT "TITLE";D$
900 LPRINT D$:INPUT "DATE (MM/DD/YY)";D$
910 LPRINT D$:INPUT "OPERATOR";D$:LPRINT "OPERATOR: ";D$
920 LPRINT "I","X","F(X)","I","X","F(X)"
930 FOR I=1 TO N STEP 2:LPRINT I,X(I),F(I),I+1, X(I+1),F(I+1):NEXT
940 B=(SX#*SY#-N*SC#)/ (SX#*SX#-N*S2#)
950 A=(SY#-B*SX#)/N
960 R#=(N*SC#-SX#*SY#)/SQR((N*S2#-SX#*SX#)*(N*SA#-SY#*SY#))
970 XB=SX#/N
980 HOME:PRINT:PRINT:
    PRINT"The calculated linear regression equation is: ":PRINT:
990 HTAB 25:PRINT " F(X)=";A;"+";B;"*  X  "
1000 PRINT:PRINT "CORRELATION COEFFICIENT: ";R#
1010 IF LEFT$(NO$,1)="P" OR LEFT$(NO$,1)="P" THEN
     LPRINT" F(X)=";A;"+";B;"* X ":LPRINT :
     LPRINT "CORRELATION COEFFICIENT: ";R#:LPRINT CHR$(12)
1020 PRINT:HTAB 34:PRINT "ACTION MENU":PRINT
1030 HTAB 14:PRINT"(P)lot  (F)orecast  (N)ew run  (O)utput
     (M)enu  (E)nd":PRINT
1032 HTAB 30:INPUT "Enter your choice ";AA$
1040 IF LEFT$(AA$,1)="P" OR LEFT$(AA$,1)="P" THEN PRINT:
     HTAB 10:PRINT"PLOTTING OPTIONS ARE:":PRINT:HTAB 12:
     PRINT"(1) REGRESSION LINE F(X) = A + BX":HTAB 12:
     PRINT"(2) DATA points only":PRINT:HTAB 25:
     INPUT "Enter your choice ( 1/2 ) ";H
1050 IF (LEFT$(AA$,1)="P" OR LEFT$(AA$,1)="P") AND H=2 THEN 1120
1060 IF LEFT$(AA$,1)="E" OR LEFT$(AA$,1)="E" THEN END
1070 IF LEFT$(AA$,1)="O" OR LEFT$(AA$,1)="O" THEN 440
1080 IF LEFT$(AA$,1)="N" OR LEFT$(AA$,1)="N" THEN 30
1090 IF (LEFT$(AA$,1)="P" OR LEFT$(AA$,1)="p") AND H=1 THEN PRINT:
     HTAB 10:INPUT "DESIRED SIGMA LIMITS (1,2,3)= ";L:
     PRINT:INPUT "ANALYSIS TYPE: (B)oth  (C)onfidence interval
     (P)rediction interval :";CP$:GOTO 1120
1100 IF LEFT$(AA$,1)="F" OR LEFT$(AA$,1)="F" THEN 1720
1110 GOTO 1030
1120 INPUT "(L)ow point or (O)rigin  ";PL$:SX=BX:SY=BY
1130 IF LEFT$(PL$,1)="O" OR LEFT$(PL$,1)="O" THEN SX=0:SY=0
1140 INPUT "(H)igh point or (P)rojection ";HP$:HX=TX:HY=TY
1150 IF LEFT$(HP$,1)="P" OR LEFT$(HP$,1)="P" THEN
     INPUT"PROJECTED X";HX:HY=A+B*HX:IF TY>HY THEN HY=TY
1160 IF LEFT$(HP$,1)="P" OR LEFT$(HP$,1)="P" THEN HY=A+B*SX:
     IF TY>HY THEN HY=TY
1170 HOME
1172 HOME:EIGHT=80:WIDTH EIGHT:
1174 PRINT:PRINT:PRINT:PRINT:PRINT:
     PRINT TAB(12);"INITIALIZING THE VIDEO SCREEN....PLEASE WAIT ....":
1176 FOR COL= 0 TO 80: FOR ROW = 0 TO 22: VIDEO%(COL,ROW)=32:
     NEXT ROW:NEXT COL
1178 PRINT:PRINT:HTAB 10:
     PRINT"**********     INITIALIZATION PROCESS COMPLETED   **********"
1180 FOR JJJ= 1 TO 400:NEXT JJJ:HOME
```

```
1182 GOTO 1190
1185 ROW=ROW + 1:IF SR$="" THEN RETURN ELSE
     FOR COL = SSS TO SSS +LEN(SR$)-1:CXL=COL-SSS+1:
     VIDEO%(COL,ROW)=ASC(MID$(SR$,CXL,1)):NEXT COL:RETURN
1190 SR$=STR$(HY):SSS=0:ROW=0:GOSUB 1185
1191 SR$="SIGMA LIMIT USED ":SSS=20:ROW =0:GOSUB 1185
1192 SR$=STR$(L):SSS=39:ROW=0:GOSUB 1185
1210 SR$=STR$((HY+SY)/2):ROW=10:SSS=0:GOSUB 1185
1230 SR$=STR$(SY):ROW=20:SSS=0:GOSUB 1185
1240 TXX=INT(100*HX)/100
1245 TB=20:TT=45:IF EIGHT=80 THEN TB=31:TT=69
1250 SR$=STR$(SX):ROW=21:SSS=0:GOSUB 1185
1251 SR$=STR$((TXX+SX)/2):ROW=21:SSS=TB:GOSUB 1185
1252 SR$=STR$(TXX):ROW=21:SSS=TT:GOSUB 1185
1260 REM PLOT THE DATA POINTS
1270 INC=(HX-SX)/60:IF EIGHT=80 THEN INC=(HX-SX)/120
1290 I=0
1300 I=I+1
1310 HIGH=(F(I)-SY)/(HY-SY)
1315 CUT=40:IF EIGHT=80 THEN CUT=60
1320 RIGHT=(X(I)-SX)/(HX-SX)*CUT
1340 ROW=21-19*HIGH:COL=RIGHT+1
1345 VIDEO%(COL,ROW)=42
1350 IF I<N THEN 1300
1360 IF H=1 THEN 1390
1362 IF LEFT$(NO$,1)="P" OR LEFT$(NO$,1)="p" THEN POKE 3,150
1364 GOTO 1375
1365 HOME: FOR ROW=0 TO 22: FOR COL=0 TO EIGHT-1
1366 PRINT CHR$(VIDEO%(COL,ROW));:NEXT COL:PRINT:NEXT ROW
1367 RETURN
1370 IF LEFT$(NO$,1)="P" OR LEFT$(NO$,1)="p" THEN POKE 3,150
1375 GOSUB 1365:POKE 3,149:HTAB 25:
     INPUT "PRESS <CR> TO CONTINUE";DZAP$:GOTO 980
1390 REM TO DRAW REGRESSION LINE
1400 FOR J=SX TO HX STEP INC
1410 Y=J*B+A
1420 HIGH=(Y-SY)/(HY-SY)
1430 RIGHT=INT((J-SX)/(HX-SX)*CUT+1)
1440 HIGH = INT(21-19*HIGH)
1445 IF HIGH < 0 OR HIGH > 22 OR RIGHT < 0 OR RIGHT > 80 THEN 1460
1450 VIDEO%(RIGHT,HIGH)=120
1460 NEXT J
1470 IF LEFT$(CP$,1)="P" OR LEFT$(CP$,1)="P" THEN 1600
1480 FOR X=SX TO HX STEP INC
1490  LX=L*SE*SQR(1/N+(N*(X-XB)^2)/ZX):HIGH=A+B*X+LX
1500 RIGHT=(X-SX)/(HX-SX)*CUT+1
1510 LOW=A+B*X-LX:
1520 HIGH=(HIGH-SY)/(HY-SY)
1530 LOW=(LOW-SY)/(HY-SY):
1540 HIGH=21-19*HIGH
1545 LOW=21-19*LOW
1546 IF HIGH < 0 OR HIGH > 22 OR RIGHT < 0 OR RIGHT > 80 THEN 1555
1548 RIGHT = INT(RIGHT):HIGH=INT(HIGH)
1550 VIDEO%(RIGHT,HIGH)=95
1555 IF LOW < 0 OR LOW > 22 OR RIGHT <0 OR RIGHT > 80 THEN 1580
```

```
1557 RIGHT= INT(RIGHT):LOW=INT(LOW):
1560 VIDEO%(RIGHT,LOW)=95:
1580 NEXT X
1590 IF LEFT$(CP$,1)="C" OR LEFT$(CP$,1)="c" THEN 1370
1600 FOR X=SX TO HX STEP INC
1610 LL=L*SE*SQR(1+1/N+(N*(X-XB)^2)/ZX):HIGH=A+B*X+LL
1620 LOW=A+B*X-LL
1630  RIGHT =(X-SX)/(HX-SX)*CUT+1
1640 HIGH=(HIGH-SY)/(HY-SY)
1650 LOW=(LOW-SY)/(HY-SY)
1660 HIGH =21-19*HIGH
1665 LOW=21-19*LOW
1666 IF RIGHT< 0 OR RIGHT > 80 THEN 1700
1667 IF HIGH > 22 OR HIGH < 0 THEN 1675
1668 RIGHT=INT(RIGHT):HIGH=INT(HIGH)
1670 VIDEO%(RIGHT,HIGH)=95
1675 IF LOW > 22 OR LOW < 0 THEN 1700
1677 RIGHT = INT(RIGHT):LOW=INT(LOW)
1680 VIDEO%(RIGHT,LOW)=95
1700 NEXT
1710 GOTO 1370
1720 WIDTH 80:PRINT:HTAB 25:PRINT"FORECAST OPTION DATA INPUT":
     PRINT:PRINT:
1722 PRINT"Input the desired X value for prediction and the desired
     prediction":PRINT"limits (1/2/3)":PRINT:PRINT:
1724 PRINT TAB(20);"PREDICTED X VALUE";:PRINT TAB(42);
     "PREDICTION LIMITS (1/2/3)   0 0 TO STOP":
     PRINT TAB(20);"----------------";:
     PRINT TAB(42);"------------------------":PRINT
1726 PRINT TAB(23);:INPUT ;X:PRINT TAB(46);:INPUT ;L:
     IF X=0 AND L=0 THEN 980
1730 LX=L*SE*SQR(1/N+(N*(X-XB)^2)/ZX)
1740 LL=L*SE*SQR(1+1/N+(N*(X-XB)^2)/ZX)
1750 F=A+B*X: FF=F-LL: FX=F+LL: FA=F-LX:FB=F+LX
1760 PRINT:PRINT:PRINT "F(X)=";F:PRINT "LOWER CONFIDENCE LIMIT= ";
     FA,"UPPER CONFIDENCE LIMIT= ";FB:
1765 PRINT "LOWER PREDICTION LIMIT= ";FF,"UPPER PREDICTION LIMIT= ";FX
1770 IF X<BX OR X>TX THEN PRINT"BEYOND RANGE OF DATA ********"
     ELSE PRINT
1780 IF LEFT$(NO$,1)="P" OR LEFT$(NO$,1)="P" THEN
     LPRINT"  X =";X,"F(X)=B+A*X=";F:
     LPRINT "Lower Confidence Limit = ";FA;
     "Upper Confidence Limit = ";FB:
     LPRINT "Lower Prediction Limit = ";FF;
     " Upper Prediction Limit = ";FX;
1790 IF LEFT$(NO$,1)="P" OR LEFT$(NO$,1)="p"
     THEN IF X<BX OR X>TX THEN
     LPRINT "BEYOND RANGE OF DATA ***********" ELSE LPRINT" "
1800 GOTO 1720
1890 DATA 1,100,2,345,13,949,10,1106,6,434,5,659,8,988,14,
     1360,9999,9999
2000 REM INITIAL PROMO SCREEN
2010 HOME:VTAB 5:HTAB 21:PRINT"*** APPLE CP/M REGRESSION ANALYSIS ***"
2020 VTAB 9:HTAB 39:PRINT"by":VTAB 14:
2030 HTAB 25:PRINT"STEVEN M. ZIMMERMAN PH.D.":
```

```
2040 HTAB 31:PRINT"LEO CONRAD"
2050 HTAB 25:PRINT"DONALD R. SMITH PH.D., P.E."
2060 VTAB 20:HTAB 25:INPUT "Press < CR > to Begin";DU$:RETURN
2300 REM DATA INPUT SUBROUTINE FOR X AND F(X)
2305 LINECOUNT%=0
2310 GOSUB 2500:
2320 HTAB 16:PRINT I;:PRINT TAB(31);:INPUT ;X(I):PRINT TAB(44);:
     INPUT ;F(I):PRINT:
2330 LINECOUNT%=LINECOUNT%+1:IF LINECOUNT%= 15 THEN
     LINECOUNT%=0:GOSUB 2700:
2340 IF F(I) <>9999 AND X(I)<>9999 THEN I=I+1:GOTO 2320
2350 RETURN
2500 REM SUB FOR DATA INPUT COL HEADINGS
2510 VTAB 2:HTAB 25:PRINT"**** DATA INPUT SCREEN ****":PRINT:
     PRINT:PRINT:
2520 HTAB 58:PRINT"9999 <CR> 9999 <CR>"
2530 HTAB 13:PRINT "ELEMENT NO.";TAB(31);"X-VALUE";TAB(42);
     "FUNCTION OF X";TAB(65);"TO STOP":
2540 HTAB 11:PRINT"-------------";TAB(28);"------------";
     TAB(41);"----------------PRINT
2550 RETURN
2700 PRINT:HTAB 58:PRINT"9999 <CR> 9999 <CR>"
2710 HTAB 13:PRINT"ELEMENT NO.";TAB(31);"X-VALUE";TAB(42);
     "FUNCTION OF X";TAB(65);"TO STOP":
2720 HTAB 11:PRINT"-------------";TAB(28);"------------";
     TAB(41);"----------------PRINT
2730 RETURN

^@
```

# Index


Accelerated Cost Recovery System (ACRS), 135, 146
Accelerated depreciation
  declining balance, 137
  sum of year's digits, 158–160
Accounting programs, 39–40
  balancing cash accounts using, 115–116
CHART.BAS (setup chart of accounts), 45–67
  checkwriting, 106–114
  double entry accounting for, 37–38
  initiating, 35–37
  INVOICE, 87–101
  mailing label, 100–105
  REPORT, 80–86
  transaction, 68–79
Account numbers, 46–47
Accounts payable, posting checks to, 108
Accounts receivable transaction files, 87
Adjustment file, 115
Adjustments (in reconciling bank accounts), 115
Advanced Logic Systems (ALS), 20
Amortization of loans, 123–129
APDOS command, 27
Apple CP/M 2.2 operating system, 20
Apple CP/M operating system, 7, 20–21
  commands, 27–30
  disk storage and, 23–26
  file names, 30–31
  starting, 22–23, 26–27
Apple-DOS (Disk Operating System), 7, 20, 22
  disk formatting by, 25
Apple II computers, 3–6
  hardware of, 21–22
  peripherals for, 7–9
  software for, 6–7
  turning on, 9
Apple IIe computers, 3
Apple II Plus computers, 3, 4
Applesoft BASIC, 5
ASM command, 27–28
Assets, 46
  declining balance depreciation for, 135–138
  depreciation of personal property under 1981 Tax Act, 155–157
  future value of, 139–140
  mortgage calculations for, 141–143
  real estate property, depreciation of, under Tax Reform Act (1981), 146–149
  straight line depreciation of, 150–151
  sum of year's digits depreciation for, 158–160
Audit trails, in transaction files, 76
Autostart ROM, 5, 9

Backup disks, 17
.BAK file extent, 31
Balance, on balloon loans, 133–134
Balance sheets, 38
  .BAL files for, 48
  manipulated by REPORT program, 81–82
  printed by REPORT program, 82–84
Balance utility, 115–116
.BAL file extent, 81
Balloon payments, 123
  call balloon note value program for, 133–134
  in cash flow analysis, 167
  in discounted value of mortgages, 153
Bank accounts, reconciling, 115
.BAS file extent, 31

Basic disk operating system (BDOS), 21
Basic input/output system (BIOS), 21
BASIC (language)
  stored in ROM, 5
  *see also* MBASIC
.BAT file extent, 31
Beginning-of-file markers, 65
Bills, INVOICE program for, 87
Binary code, 23
Bits, 3
Bonds, discounted value of, 152–154
Book value, 135, 151
Bootstrap programs, 22, 27
Break-even analysis, 130–132
Browsing, in INVOICE program 96–97
Bytes, 3

Call balloon note value program, 133–134
  mortgage calculations program branching to, 141
Cash accounts, balancing, using transaction program, 115–116
Cash flows
  analysis of, 119, 161–171
  code of program for analysis of, 187–194
  in discounted value of bonds and mortgages, 152
  in present value of loans, 154
  unequal, present value of, 144–145
CCUST$ file, 99
Central processing units (CPUs), 20–21
  Synertek 6502, 3, 5, 22
Characters, binary code for, 23
Charts of accounts
  inputting files for, 52–63
  numbering system for, 46–47
  numbers in, 63–64
  posting invoices on, 94
  program for, 63–67
  program to setup (CHART.BAS), 45
  running program, 47–52
Checks, 106
  uncleared, 115
Checkwriting program, 106–109, 112–114
  code of, 195–200
  keyboard input to, 110–111
  supplier's file used with, 109–110
  transaction files used with, 112
CLEAR statement (BASIC), 64–65
CLOSE statement (BASIC), 66
.COM file extent, 30, 31
Commands
  Apple CP/M, 27–30
  MBASIC, 36–37
COMPANY.DAT file, 87, 88, 99
Compound interest
  in calculating future value of assets, 139–140
  *see also* Interest
Computers
  hardware of, 21–22
  setting up, 9
Confidence limits, in regression analysis, 180–181, 183–184
CONFIGIO command, 28
Console command processor (CCP), 21
Control program, 27
COPY command, 13, 16, 28
Copying programs
  on dual-drive systems, 12–15
  on single-drive systems, 15–18
Correlation coefficients, 180, 184
Costs
  in break-even analysis, 130
  in cash flow analysis, 164
  in real estate property depreciation, 146
CP/M 2.2 operating system, 20
CP/M 3.0 operating system, 20
CPM56 utility, 29
CP/M+ Card (Advanced Logic Systems), 20
CP/M operating system
  Apple CP/M, 7
  copying diskettes under, 12–18
  disk formatting by, 25

*see also* Apple CP/M operating system
Credit balances, 46
Credits, 37
in charts of accounts, 59
in transaction files, 71
Current assets, 46
Customer files, 87
in INVOICE program, 88
mailing label program handling of, 103–104
Customer numbers, 99–100
Cut sheet feeders, 8–9

Daisy wheel printers, 8
Data, stored on disks, 23–26
Data files, 23
DATA statement (BASIC), 146, 149, 157
input for forecasting program from, 173–174
.DAT file extent, 31
DDT command, 28
Debits, 37
balances, 46
in charts of accounts, 59
in transaction files, 71
Declining balance depreciation, 135–138
Depreciation
in cash flow analysis, 162, 165–166
declining balance, 135–138
future value of assets for, 139–140
of personal property under 1981 Tax Act, 155–157
of real estate property, under Tax Reform Act (1981), 146–149
straight line, 150–151
sum of year's digits, 158–160
unequal cash flows and, 144
Dictionary programs, 9
Digital Research, 21, 27
DIM statement (BASIC), 65
DIR command, 13, 14, 18
Discounted value of bonds and mortgages, 152–154
Disk directory, 26
Disk drives, designations of, 9
Diskettes
copying, on dual-drive systems, 12–15
copying, on single-drive systems, 15–18
data storage on, 23–26
formatting of, by operating systems, 22–23
Disk operating systems (DOS), 7, 20–21
Autostart ROM for, 5
*see also* Apple CP/M operating system; CP/M operating system
Disk sectors, 25
Disk tracks, 25
.$$$ file extent, 31
Double entry accounting, 37–38
DOWNLOAD command, 28
DUMP command, 28
Dynamic Debugging Tool (DDT), 28

Economic Recovery Act (1981), *see* Tax Reform Act (1981)
ED command, 28
Electricity, surge protection for, 9
End-of-file markers, 65
for customer files, 88
in supplier's file, 110
Envelopes, printed by mailing label program, 105
Epson MX-80 printer, 11
Epson MX-100 printer, 11
Equal payment loans, 124–126
call balloon note value program for, 133–134
in cash flow analysis, 167
interest calculations for, 128
Equal reduction loans, 124, 126–127
in cash flow analysis, 167
interest calculations for, 128
ERA command, 30
Errors
in cash accounts, 116
rounding, 124–126, 133–134, 142, 173

EVALPAC, 119–122
break-even analysis in, 130–132
call balloon note value program, 133–134
cash flow analysis program, 161–171
declining balance depreciation program, 135–138
depreciation of personal property under 1981 Tax Act program, 155–157
discounted value of bonds and mortgages program, 152–154
forecasting program, 172–186
future value of assets program, 139–140
loan amortization program, 123–129
mortgage calculations program, 141–143
present value of unequal cash flows program, 144–145
real estate property depreciation program, 146–149
straight line depreciation program, 150–151
sum of year's digits depreciation program, 158–160
.EXE file extent, 31
Expansion slots, 3
Extents (in file names), 30–31

Fields, 23
Files, 23
in accounting programs, 40
created in CHART.BAS program, 48
initializing, 45
input for forecasting program from, 173
mailing label program handling of, 103–104
names for, 30–31
open, in MBASIC, 39
opened by REPORT program, 80
used by checkwriting program, 109–110
used by INVOICE program, 87–89, 95, 99
used by transaction program, 68–73
FILES command (MBASIC), 36
Financial analysis
break-even analysis, 130–132
cash flow analysis, 161–171
discounted value of bonds and mortgages program for, 152–154
forecasting program for, 172-186
future value of assets in, 139-140
mortgage calculations for, 141–143
Financial statements, 37–38
Fixed assets, 46
Fixed costs, 130
Floppy diskettes, 23–25 *see also* Diskettes
Forecasting, future value of assets for, 139–140
Forecasting program, 172–182, 184–186
code of, 211–216
theory behind, 182–184
FORMAT command, 12, 15, 28
Formatting, by operating systems, 22, 25
Forms, pre-printed checks, 106
used with INVOICE program, 97, 100
Full disk operating system (FDOS), 21
Furniture, 9
Future value of assets, 139–140

GBASIC, 29
General ledger program, 37–38
General ledger systems, 68
Graphics
in forecasting program, 172, 181–182
printing of, 11
Graphtrax, 11

Hard sectored diskettes, 25
Hardware, 21–22
of Apple II computers, 3–5

peripherals, 5–9
.HEX file extent, 31

.INC file extent, 45, 81
Income, in cash flow analysis, 164
Income and expense statements, 37–38
.INC files for, 48
inputting information for, 55–58
manipulated by REPORT program, 81–83
posting invoices on, 94
Index hole, 25
Inflation, in cash flow analysis, 162, 168, 170
Integer BASIC, 5
Interest
in calculating future value of assets, 139–140
in discounted value of bonds and mortgages, 152
on loans, 124
theory on calculation of, 127–128
in time value of money, 145
Interest only loans, 124, 127
in cash flow analysis, 167
interest calculations for, 128
Internal rate of return, 162, 168
Investments, cash flow analysis of, 161–171
Investment tax credits, 155
INVNUM.DAT file, 87, 88
INVOICE program, 87–101
code of, 217–226
I/O Byte, 132

Journals, 76

Labels, program for generation of, 100–105
Land
cash flow analysis of investments in, 163
not subject to depreciation, 158
value of, in cash flow analysis, 166
*see also* Real estate property
Least squares analysis, 172, 184
LEFT$ statement (BASIC), 65
Letter quality printers, 8
Liabilities, 46
Linear approximation techniques, 143
LOAD command, 28
Loans
amortization of, 123–129
call balloon note value program for, 133–134
in cash flow analysis, 167
mortgages, 141–143
mortgages, discounted value of, 152–154
Long term liabilities, 46
Losses, in charts of accounts, 59
Low income housing (under 1981 Tax Reform Act), 147
in cash flow analysis, 166

Mailing label program, 100–105
code of, 228–233
.MAP file extent, 31
Markers
beginning-of-file and end-of-file, 65
for customer files, 88
in supplier's file, 110
Master disks, 16
MBASIC
commands, 36–37
copying, 13–14, 17–18
files open in, 39
MBASIC.COM file, 17, 18
MBASIC 5.0, 20, 22
MBASIC command, 28–29
Memory, 4–5
addressed by CPU, 21
MENU program, 35, 41–44
code of, 227
REPORT program not accessed from, 80
Menus
in checkwriting program, 107
in EVALPAC, 119–121
in INVOICE program, 87, 90
in mailing label program, 103
in transaction program, 71–72

Merging
of files, 60–61
by forecasting program, 175, 177–178
transaction files, 73
Microprocessors, 3, 5, 20
Microsoft, 20
Microsoft Z-80 Softcard, 18, 20, 22
Models, economic, 161
Money, time value of, 145
Monthly transactions, 73–77
Mortgages, 141–143
in cash flow analysis, 167
discounted value of, 152–154

Names, for files, 30–31
for transaction files, 69
NAMEX.DAT file, 87, 88
Net present value, 161
Non-volatile memory, 4
Numbers
in charts of accounts, 46–47, 63–64
for checks, 107–108
for invoices, 90–91
manipulation of, in regression analysis, 173
rounding of, 66
rounding errors and, 124–126, 133–134, 142, 173
in transaction files, 70–71

.OBJ file extent, 31
OPEN statement (BASIC), 66
Operating systems, 7, 20–21
Autostart ROM for, 5
starting, 22–23
*see also* Apple CP/M operating system; CP/M operating system

Paper, 8–9
Payments
call balloon note value program for, 133–134
in mortgage calculations, 141
Peripheral cards, 3
Peripheral Interchange Program (PIP), 14, 17, 29
Peripherals, 5–9
Personal property, depreciation of, under 1981 Tax Act, 155–157
PIP command, 14, 29
POKE statement (BASIC), 132
Posting (accounting), 37
of checks, 108
of invoices, 90
of transactions, 68
Power supply, surge protection for, 9, 26
Prediction limits, in regression analysis, 181, 184
Present value
net present value, 161
of unequal cash flows, 144–145
Printers, 11, 52
letter quality, 8
space for, 9
Printing
break-even analysis charts, 132
cash flow analysis, 168–170
charts of accounts, 51–52
checks, 109
graphics output from forecasting program, 181–182
invoices, 90, 93–94
reports, 82–84
transaction files, 76–77
Profit and loss statements, *see* Income and expense statements
Profits, in charts of accounts, 59
Programs, 6
accounting, 35–40
break-even analysis in, 130–132
call balloon note value, 133–134
cash flow analysis, 161–171, 187–194
CHART.BAS (setup chart of accounts), 45–67
checkwriting, 106–114, 195–200
copying, on dual-drive systems, 12–15
copying, on single-drive systems, 15–18

declining balance depreciation, 135–138
depreciation of personal property under 1981 Tax Act, 155–157
discounted value of bonds and mortgages, 152–154
EVALPAC, 119–122
forecasting, 172–186, 211–216
future value of assets, 139–140
INVOICE, 87–101, 217–226
loan amortization, 123–129
mailing label, 100–105, 228–233
MENU, 41–44, 227
mortgage calculations, 141–143
present value of unequal cash flows, 144–145
real estate property depreciation, 146–149
REPORT, 80–86, 234–237
SETUP, 238–242
straight line depreciation, 150–151
sum of year's digits depreciation, 158–160
transaction, 68–79, 115–116, 243–247
Prompts, 27
Property
personal, depreciation of, under 1981 Tax Act, 155–157
*see also* Real estate property
Purchase order numbers, 92

Random access memory (RAM), 4
addressed by CPU, 21
Rates of return
of bonds and mortgages, 152–153
internal, 162, 168
Read only memory (ROM), 4–5
Real estate property
cash flow analysis of investment in, 161–171
depreciation of, according to Tax Reform Act (1981), 146–149
mortgage calculations for, 141–143
sum of year's digits depreciation of, 158
Records, 23
Regression analysis, 119, 173, 180–182
equation for, 179
theory behind, 182–184
REM statements (BASIC), 42
REN command, 30
REPORT program, 80–86
code of, 234–237
Reports, of balance sheets and income expense statements, 62
RESTORE statement (BASIC), 149, 157
Revenues, in break-even analysis, 130
ROM (read only memory), 4–5
Rounding errors, 124–126
in call balloon note value program, 133–134
in forecasting program, 173
in mortgage calculations program, 142
Rounding of numbers, 66
Rule of 78, 159
RW13 utility, 29

Sales taxes, 92
Sales terms, 93, 100
Salvage value
in cash flow analysis, 166
in declining balance depreciation, 135
in straight line depreciation, 150
in sum of year's digits depreciation, 158
Saving
data from forecasting program, 176
programs, 42
Searching
by mailing label program, 105
of supplier's file, 111
Sectors (disk), 25
Selling terms, 93, 100
Setup chart of accounts (CHART.BAS) program, 45
SETUP program, 238–242
Short term liabilities, 46
Simple interest, 124
Slave disks, 16
Soft sectored diskettes, 25

Software
  for Apple II computers, 6–7
  operating systems, 20–21
Sorting, by mailing label program, 104
Spelling checkers, 9
Spindle hole (on diskettes), 24
STAT command, 29
Stockholders' equity, 46
Straight line depreciation, 150–151
SUBMIT command, 29
Sum or year's digits depreciation, 158–160
Supplier's files, 109–111
Surge protection, 9, 26
Synertek 6502 microprocessor chip, 3, 5, 22
System parameter area, 21

Tax Act (1982), 155
Taxes
  in cash flow analysis, 162, 164, 166, 167
  declining balance depreciation for, 135–138
  depreciation of personal property under 1981 Tax Act for, 155–157
  depreciation of real estate property, under Tax Reform Act (1981) for, 146–149
  sales, in invoices, 92
  sum of year's digits depreciation for, 158–160
Tax Reform Act (1981), 155, 162, 166
  declining balance depreciation ended by, 135
  real estate depreciation under, 146–149
  sum of year's digits depreciation ended by, 158
Terms of sales, 93, 100
Time value of money, 145
Title accounts, 47
Titles, on balance sheets and income and expense statements, 62
Tracks (disk), 25
Transaction files, 39
  for accounts receivable, 87
  manipulated by REPORT program, 81
  used by checkwriting program, 106, 107, 112
  used by INVOICE program, 90
Transaction program, 68, 77–79
  to balance cash accounts, 115–116
  code of, 243–247
  monthly runs using, 73–77
  reusable files in, 68–73
Transactions, 68
  balancing, 115–116
Transient program area (TPA), 21

Unequal cash flows, present value of, 144–145
Utilities, 27–30
  PIP, 14, 17

Values
  discounted, of bonds and mortgages, 152–154
  net present value, 161
  time value of money, 145
Variable costs, 130
Variables
  in charts of accounts, 48–49
  in forcasting program, 186
Volatile memory, 4
Volumes, 23

WIDTH statement (BASIC), 131
Word processing, options for, 8
Write-protect notch, 24–25

XSUB command, 29

Z-80 cards, 18, 22
Zeroing, of charts of accounts, 59
Zilog Z-80 microprocessor, 21, 22

# *Business Applications For The Apple II and IIe Under CP/M*

**Steven M. Zimmerman, PhD, Leo M. Conrad, and Donald Smith, PhD, P.E.**

Put your Apple to work for you! With the help of this comprehensive collection of programs, written in MBASIC, you'll have your business up and running in no time! Includes programs for using your Apple to learn the day-to-day applications and fundamentals of finance and accounting. Presents step-by-step information on how to customize these programs to your specific needs. What's more—if your're a first-time Apple user, there's a discussion of Apple and CP/M!

Inside this book, you'll find programs to help you:

- Produce balance sheets and profit and loss statements
- Write checks and invoices
- Compile general ledger results
- Determine the future value of assets
- Create depreciation tables
- Perform complete analyses of real estate investments
- Analyze and evaluate the financial health of your business

Plus—there are many more programs tailor-made to help you and your Apple (operating with CP/M) breeze through complex calculations!

## CONTENTS

The Apple II Microcomputer System with CP/M • Apple CP/M • A Collection of Business Programs • (S)etup Chart of Accounts • (T)ransaction Program • Report:Income/Expense and Balance Sheet • Invoice Program • Mailing Label Program • Checkwriting Program • Using TRANsaction to Balance Your Cash Account • Statistical and Analysis Menu • Loan Amortization • Break-Even Analysis • Call Balloon Note Value • Declining Balance Depreciation • Future Value of an Asset • Mortgage Calculations • Present Value of Unequal Cash Flows • Real Estate Property-1981 Depreciation • Straight Line Depreciation • Value Discounted of a Bond or Mortgage • Depreciation Personal Property 1981-Law with 1982 Revisions • Sum of the Year's Digits Depreciation • Cash Flow Analysis • Forecast • Index

ISBN 0-89303-354-5

Zimmerman
Conrad
Smith

# BUSINESS APPLICATIONS for the Apple II&IIe under CP/M

Brady